Oracle® Enterprise Communications Broker

Administrator's Guide
Release P-CZ2.2.0

September 2017

Notices

Contents

About This Guide

The *Oracle® Enterprise Communications Broker Administrator's Guide* provides the following information about the Oracle Enterprise Communications Broker (ECB) hardware and software.

- Supported platforms
- How to get the system operational
- Initial configuration
- Maintenance and troubleshooting

Related Documentation

The following table describes the documentation set for the ECB.

Document Name	Document Description
Administrator's Guide	Describes how to deploy the system.
Embedded Help system	Contains task-oriented topics for configuring, administering, maintaining, and troubleshooting the ECB hardware and software.
Release Notes	Contains information about the current release, including specifications, requirements, new features, enhancements, inherited features, known issues, caveats, and limitations.
SBC Family Security Guide	Provides information about security considerations and best practices from a network and application security perspective for the Enterprise family of products.
User's Guide	Describes how to configure SIP signaling management and how to tailor the system to specific needs.

Revision History

The following table describes updates to this guide.

Date	Description
July 2017	• Initial Release
September 2017	• Updates the Minimum VM Resources topic.

1

Applicable Platforms

The Oracle Enterprise Communications Broker is available either as an appliance or as a application for operation on virtual machines. When running as an appliance, the Oracle Enterprise Communications Broker software is packaged with the Netra Server X3-2 for Acme Packet and delivered to the end customers. When running as a virtual application, the Oracle Enterprise Communications Broker software can be deployed on any third-party COTS hardware that meets the specified guidelines.

When delivered as an appliance, the application comes pre-installed on Oracle's Netra Server X3-2 for Acme Packet. Server cabling instructions, which also identifies key hardware elements, such as interfaces, are presented below. Instructions on installation and maintenance of the Netra Server X3-2 for Acme Packet are generic to SBC, Session Router and other appliance applications.

The generic Netra Server X3-2 for Acme Packet documentation herein identifies all hardware interfaces. With respect to cabling the Oracle Enterprise Communications Broker, the applicable interfaces, as named in the hardware documentation, include:

- s0p0—Service access
- wancom0—Management access
- wancom1—High Availability (HA) access
- SER MGT(COM1)—Serial management access

You run the application as a virtual machine over a VM system, such as Oracle VM Server. You use VM management software, such as Oracle VM Manager, to create and maintain your virtual machines.

Virtual machine installation instructions are available in the Platforms chapter of the *Oracle Enterprise Session Border Controller Configuration Guide*. Generic hardware information is provided in the applicable documentation provided by your hardware vendor.

Software Packaging

The Release PCz version 2.2.0 build image is labeled nnPCz220.bz. The image is compressed by the zlib software library and includes all software components needed to install and operate the Oracle Enterprise Communications Broker (ECB).

Note: Note that you must obtain a license if you want to operate with TLS. The procedure to obtain this license is documented in the Administrator's Guide.

Oracle Enterprise Communications Broker software delivered for virtual machines includes the following packages:

Image Name	Description
nnPCz220.bz	Standalone compressed image - This .bz image package is primarily used to load and operate the ECB software as an appliance. You can also use the .bz image as a load image to existing virtual machines. Create your virtual machine according to specifications. Then copy this image to your machine (eg /code) and point your boot parameters to it.
nnPCz220-img-bin.ova	Virtual Machine Template - Import within virtual machine hypervisor to create the entire machine.

Netra X5-2 for Communications Software Installation

Oracle Communications produces a variety of software products that run on the Netra X5-2 for Communications platform, including Oracle session delivery applications.

Use the following steps to prepare the Netra X5-2 for Communications for session delivery software installation.

1. Confirm applicable firmware on the server.
   - To check the firmware versions installed in the server, go to the Oracle Integrated Lights Out Manager (ILOM) web interface, and navigate to **System Information** > **Firmware**.
   - Software and firmware versions certified for use with Oracle Session Delivery products include:
     - ILOM—v3.2.4.32, r96182
     - BIOS—32000013
2. Upgrade or downgrade the firmware on the server as necessary. Go to https://docs.oracle.com/cd/E37444_01/index.html for ILOM upgrade instructions.
3. Configure the BIOS settings. (Settings navigation may differ based on the BIOS version.)
   a. Observe the boot procedure and use the documented key sequence to interrupt the boot and display the BIOS configuration dialogs. For example, pressing the F2 key is a common way to enter BIOS configuration from a terminal application that supports function keys.
   b. Navigate to the Boot menu and, depending on the software distribution you are using, set the USB or CD as the first device followed by the disk controller.
   c. Disable Hyper-Threading.
   d. Disable CPU power limit.
   e. Disable C6 Reporting.
   f. Disable the UEFI Stack.
   g. Change Energy Performance to Performance. (For example, set "ENERY_PERF_BIAS_CFG" mode to "PERF".)
   h. To decrease boot up time, Oracle recommends disabling Intel PXE Boot Agent for both onboard and NIC ethernet ports. To disable the Boot Agent for the onboard ethernet ports, navigate to the OpROM option for NET0, NET1, NET2, and NET3 interfaces (for example, IO > Internal Devices) and set it to disabled.
   i. To disable Boot Agent for NIC ethernet ports, note the blue PCIe slot number label at the back of the Netra server where the NICs are installed, then disable the OpROM option for those slots. (Note that you may be able to identify slot number through the ILOM System Information > PCI Devices menu.)
   j. Reboot the server.
4. Initialize the HDD.
   a. Open the ILOM remote system console to observe the system's boot cycle, and interrupt the boot cycle to enter the LSI MegaRAID status display. For example, pressing the Ctrl-R key is a common way to enter LSI MegaRAID BIOS Configuration Utility.

- **b.** Navigate the utility to establish the elements of your virtual drive, typically consisting of a New Configuration with two entire HDDs.
- **c.** Access the menu from which you create a virtual drive.
- **d.** Set the RAID level to RAID-1.
- **e.** Select all of the drives that you want.
- **f.** From the Virtual Drive Management dialog, select the new drive and initialize it. For example, pressing F2 and selecting Fast Init from the command menu is a common way to execute initialization.
- **g.** After initialization is complete, Escape from the LSI MegaRAID Configuration Utility and reboot the system.

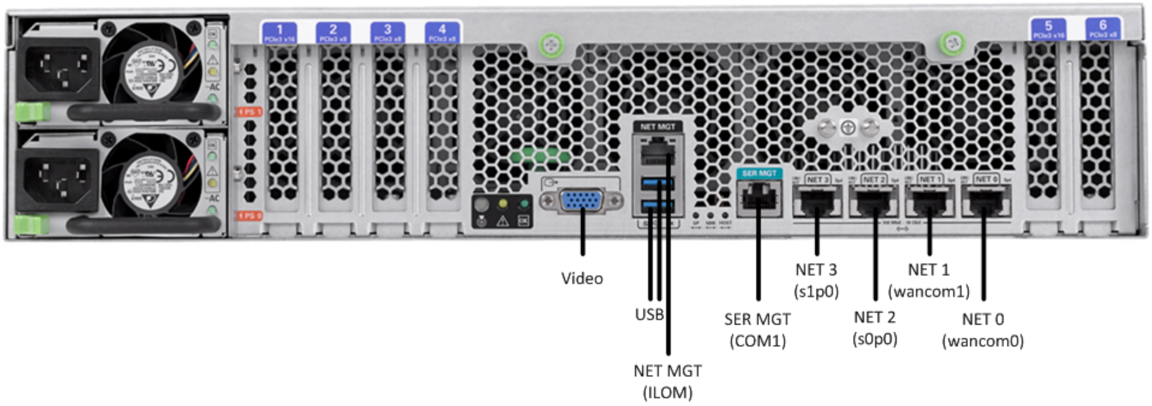
5. Perform a cold shutdown by removing all system power.

Cable the Netra X5-2 for Communications

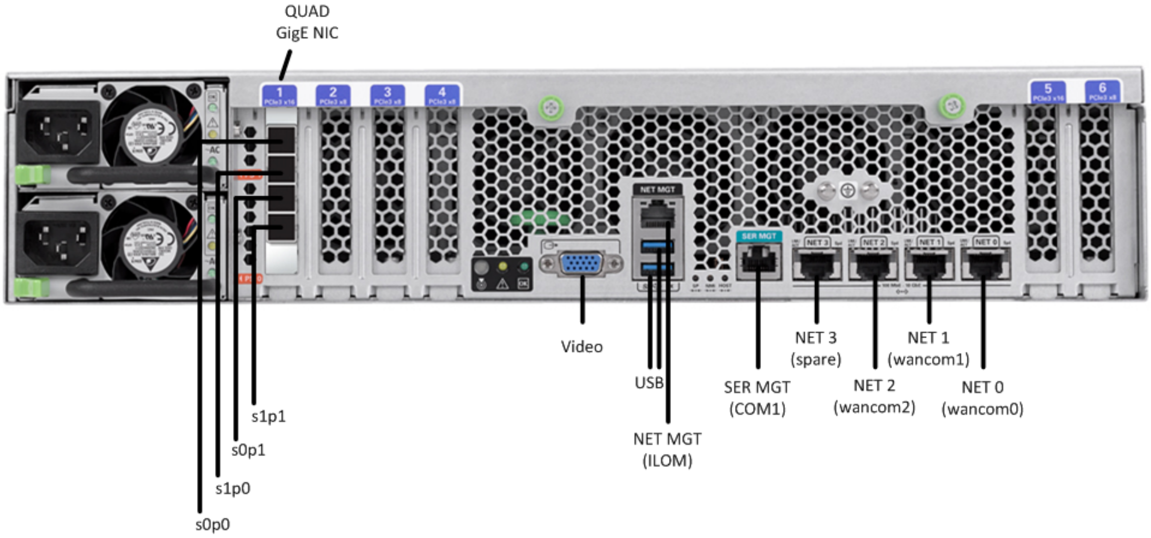
After mounting the Netra X5-2 for Communications in an equipment rack and installing all components, use the following instructions to connect all appropriate data cables to the ports before powering the system up and beginning the configuration.

Oracle supports the following configurations of the Netra X5-2 for Communications (the onboard 10 GigE ports are configured for 1G operation):

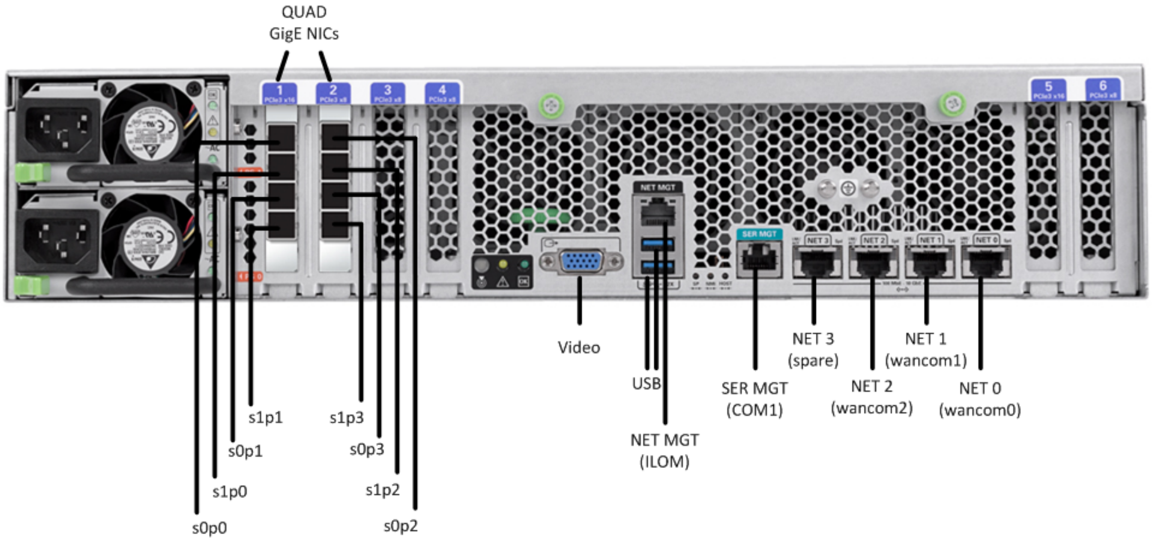
- Configuration A: Four onboard 10 GigE ports and no Quad GigE NIC
- Configuration B: Four onboard 10 GigE ports and 1 Quad GigE NIC
- Configuration C: Four onboard 10 GigE ports and 2 Quad GigE NICs



Netra X5-2 for Communications Configuration A (4 Onboard 10 GigE Ports)



Netra X5-2 for Communications Configuration B (4 Onboard 10 GigE Ports & 1 Quad GigE NIC)



Netra 5-2 for Communications Configuration C (4 Onboard 10 GigE Ports & 2 Quad GigE NICs)

Oracle recommends using Category 6 (or better) for all Ethernet connections.

You do not need to use every port for proper operation.

You can install and remove Ethernet and 1000BASE-T cables while the Netra X5-2 for Communication runs, but when you disconnect a cable the link is lost and the system generates an alarm.

Available Connections

Please read all of the information for each of the available connections prior to cabling the Netra X5-2 for Communications.

Port	Description	You Need:
NET (0-3)	10 GigE ports - labeled Net 3, Net 2, Net 1 and Net 0 (left to right). Enables you to connect the Netra X5-2 to your network.	A Category 6 (or better) Ethernet cable to connect to the NET 0 port to your network Network parameters such as an IP address (can be provided by DHCP

Port	Description	You Need:
		services or assigned a static address in the OS) Additional Category 6 (or better) Ethernet cables and Ethernet addresses as needed for additional connections to NET 1 - 3
NET MGT	Provides a 10/100BASE-T Ethernet connection to the Service Processor (SP) through an RJ-45 connector. The NET MGT port provides support connections to the SP using the Oracle Integrated Lights Out Manager (ILOM) CLI and Web interface. By default, the NET MGT port is configured to use DHCP to automatically obtain an IP address. Alternatively, you can assign a static IP address to the NET MGT port. To use the NET MGT port, you must configure its network settings. Once configured, use the NET MGT port IP address to log on to the device using a browser or secure shell.	Category 6 (or better) Ethernet cable to connect the NET MGT port to your network IP address for this port (required from DHCP or a static address)
SER MGT (COM1)	Provides a TIA/EIA-232 serial Oracle/Cisco standard connection to the SP through an RJ-45 connector. Default settings: 8N1: eight data bits, no parity, one stop bit 115200 baud Disable hardware flow control (CTS/RTS) Disable software flow control (XON/XOFF)	A terminal device (e.g., terminal, connection to a terminal server, or computer such as a laptop running terminal emulation software) A cable to connect the terminal device to the SER MGT (COM1) port
USB	Provides USB connections to the SP. The USB ports are hot pluggable, so you can connect and disconnect USB cables from these ports and peripheral devices without affecting server operations.	USB keyboard USB mouse Note: Maximum USB cable length: 5 meters
VIDEO	Provides a temporary video connection to the SP.	VGA monitor HDB-15 video cable with a maximum cable length of 6 meters (19.7 feet)

Cable the Local Console

You can connect the Administration console to either the Oracle Integrated Lights Out Manager (ILOM) (NET MGT), the local VGA+USB console ports, or the local SER MGT (COM1) serial console port.

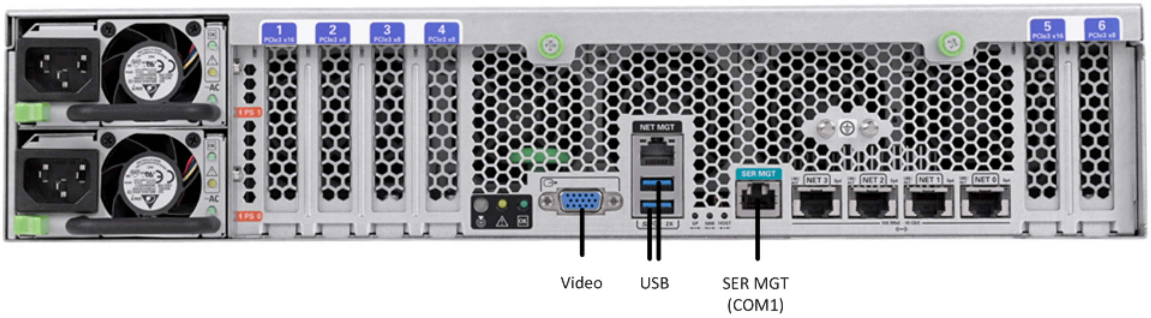
- To cable a serial console connection:
  - Serial console cable with an RJ-45 connector
- To cable a USB and Video Connection:
  - DB-15 video cable with a maximum cable length of 6 meters (19.7 feet)
  - USB cable with a maximum cable length of 6 meters (19.7 feet)
  - USB keyboard

    In the following procedure, you have the option to either cable a serial connection or to cable a USB/ Video connection.

Note: Do not configure COM2 in the bootparams menu.

When configuring boot loader parameters, set the console to VGA when you use ILOM or VGA+USB, or to COM1 when you use SER MGT. The boot loader is accessible on all console ports, but only input from the active console port can be recognized by the Netra X5-2 for Communications.

1. Locate the appropriate cables to connect to the Netra X5-2 for Communications.
2. To cable a serial connection, insert the serial console cable into the SER MGT (COM1) port.



Connecting to USB, VGA and SER MGT (COM1) Ports

Note: Refer to the Netra X5-2 for Communications hardware documentation for information on how to configure the terminal application to connect to the console, and how to establish communications with the Netra X5-2 for Communications.

3. To cable a USB/Video connection, do the following:
   a) Insert the 15-pin connector end of the video cable into the Video port.
   b) Insert the USB cable from the mouse and keyboard into the USB ports.
4. Lead the cables neatly away from the rear panel.
5. Plug in the cables to their respective destination components.

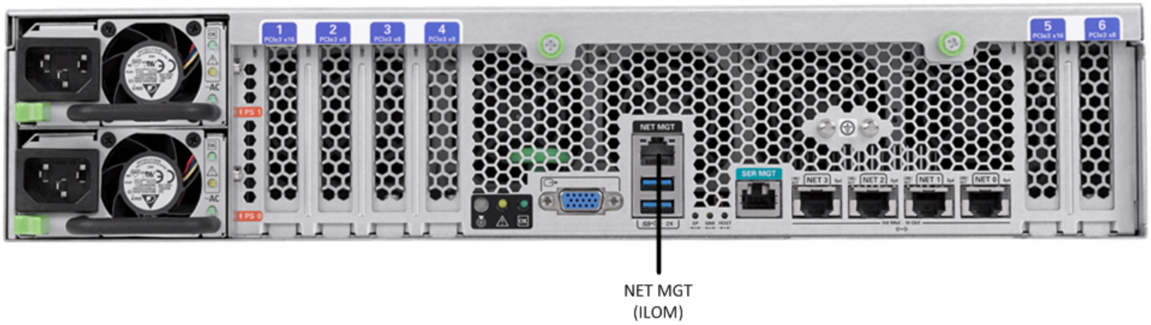
Connect ILOM to the Netra X5 for Communications

Use the following procedure to make a connection to the Netra X5-2 for Communications Oracle Integrated Lights Out Manager (ILOM) port. For a remote permanent connection to the Service Processor over the ILOM connection, use the rear panel NET MGT port.

Note: Keep Ethernet cables separated from power cables by at least 60mm where possible and never run them in the same channel of the rack without segregation.

- Category 6 (or better) Ethernet cable with RJ-45 jacks

1. Locate the cable to connect to the Netra X5-2 for Communications.
2. Plug the RJ-45 connector into the ILOM port.



3. Lead the cable neatly away from the rear panel.
4. Connect the other end of the cable to the LAN.

- Refer to the Netra X5-2 for Communications hardware documentation for information on how to configure the Web browser application to connect to the console, and how to establish communications with the Netra X5-2 for Communications.

Software Installation - Netra and Server-based Platforms

Oracle Communications Session Delivery software requires software installation when deployed on Netra and Server-based.

Install Procedure

Software installation to Netra and server-based platforms includes the following high-level steps:

1. Ensure your device is set to boot from your boot media. This may be via USB or CD.
2. Insert your installation media in any USB slot or CD drive.
3. Power up the device, observing the boot cycle.
4. When power-up is complete, the device loads the Oracle Enterprise Communications Broker software. Wait for this to complete.
5. When notified, remove the boot media and allow the device to boot to the newly installed Oracle Enterprise Communications Broker software.

   (This step may not be required as some platforms support a boot priority mechanism that knows to boot from your hard drive after the installation is complete.)

Note: Note that the Oracle Enterprise Communications Broker boots by default to VGA (or as configured by BMC) during the installation. The user can change this to serial "temporarily" during install. Within the boot parameters and after installation, however, the user can set the boot option to VGA or serial. This setting is “permanent”, meaning that any device set to boot to VGA appears “dead” at serial (and vice-versa).

Logging Into the System

Anytime the runtime image has loaded, including the first time, the ACLI login prompt appears in your terminal application as follows:

```
User Access Verification
Password:
```

If the Oracle Enterprise Communications Broker completed booting before you connected to the console port, press the <Enter> key on the console keyboard a few times to activate the console connection.

System access in the following procedure uses the default User and Superuser passwords.

1. At the ACLI Password prompt, enter the default system User password and press <return>. Your entries are not echoed on the screen.

   ```
   User Access Verification
   Password:
   ORACLE>
   ```

   From the User prompt you can view various configuration states and operating statistic, but you can not perform configuration tasks.
2. Type enable and press <return> to enter Superuser mode. The prompt to enter the superuser password appears.

   ```
   ORACLE> enable
   Password:
   ```
3. Enter the Superuser password and press <return>. The system prompt will ends with a pound sign instead of a closed-angle-bracket to let you know are in Superuser mode.

   ```
   Password:
   ORACLE#
   ```

From this point, you format your disk and set your boot parameters.

Disk format is required the first time you power on. Refer to the Oracle Enterprise Communications Broker's Configuration Guide to learn about further configuration procedures.

First Steps after Software Installation

Oracle recommends the following steps after installation on COTS/Server-based platforms.

1. Execute the Oracle Enterprise Communications Broker's **format hard-disk** command, per your requirements.
2. Go to the boot parameters to set your “Target Name” to your preferred Oracle Enterprise Communications Broker name.
3. Go to the boot parameters to set your “Console Device” to com1 (serial).

   Note: Note at the boot parameters that the default system is named “/boot/bzImage”. Be aware that upgrading code includes obtaining images with, for example, an SCz prefix and the .bz file extension.
4. Go to the boot parameters to set the "IP Address" to your preferred management port IP address.
5. By default, the system recognizes the first on-board interface as wancom interfaces and any subsequent interfaces as media interfaces. Configure your interfaces via ACLI explicitly, as desired.
6. Turn off the Oracle Enterprise Communications Broker using the **`Halt`** command. This provides you with a graceful software shutdown, after which the hardware is still powered on.
7. Power down the hardware using the power switch.

Known Issues

Netra X5-2 for Communications

The **interface-mapping locate** command does not work with the Netra X5-2 for Communications onboard interfaces. The command does work with PCI interfaces installed on the platform.

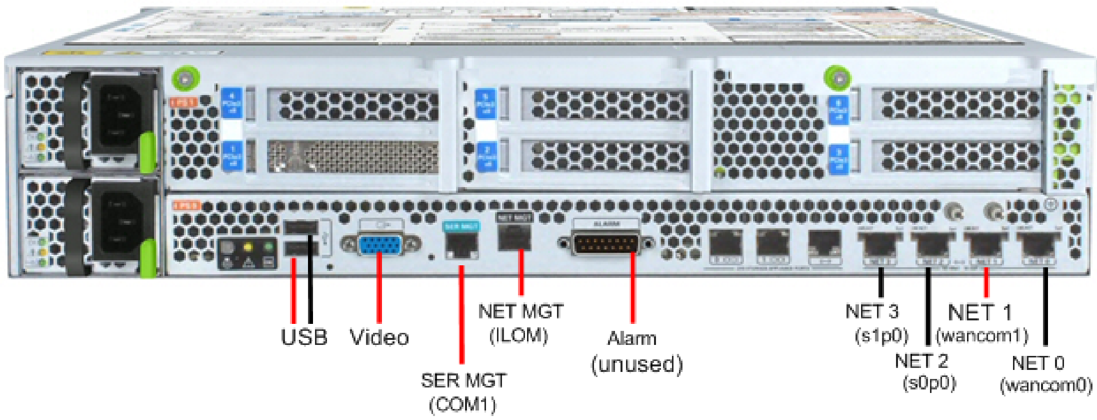
The Onboard Ethernet ports of the Oracle X-series servers (X3-2, X5-2, and so on.) run natively at 10GBASE- T, which requires the use of Category 6a cabling. These ports negotiate down to 1000BASE-T or 100BASE-T, but the negotiation might not succeed when you use incompatible cables. For example, do not use Cat5/5e cables or Cat6 cables not rated for 500MHz operation.

Cabling the Netra Server X3-2 for Acme Packet

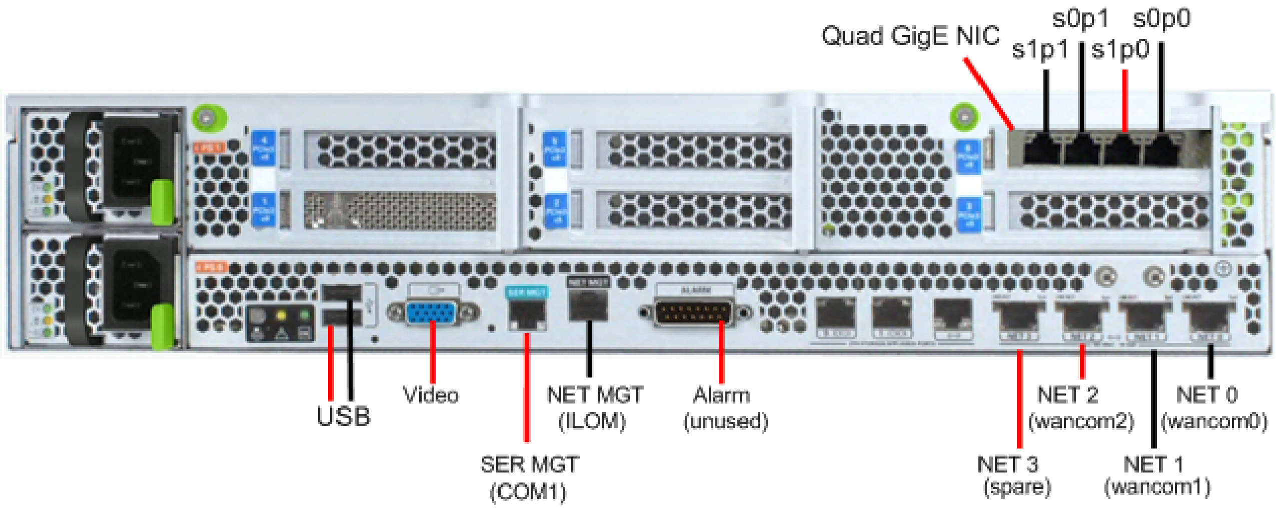
After mounting the Netra Server X3-2 for Acme Packet in an equipment rack and installing all components into it, connect all appropriate data cables to the ports before powering the system up and configuring it. This section describes how to make data cable connections.

Oracle supports the following configurations of the Netra Server X3-2 for Acme Packet (the onboard 10 GigE ports are configured for 1G operation):

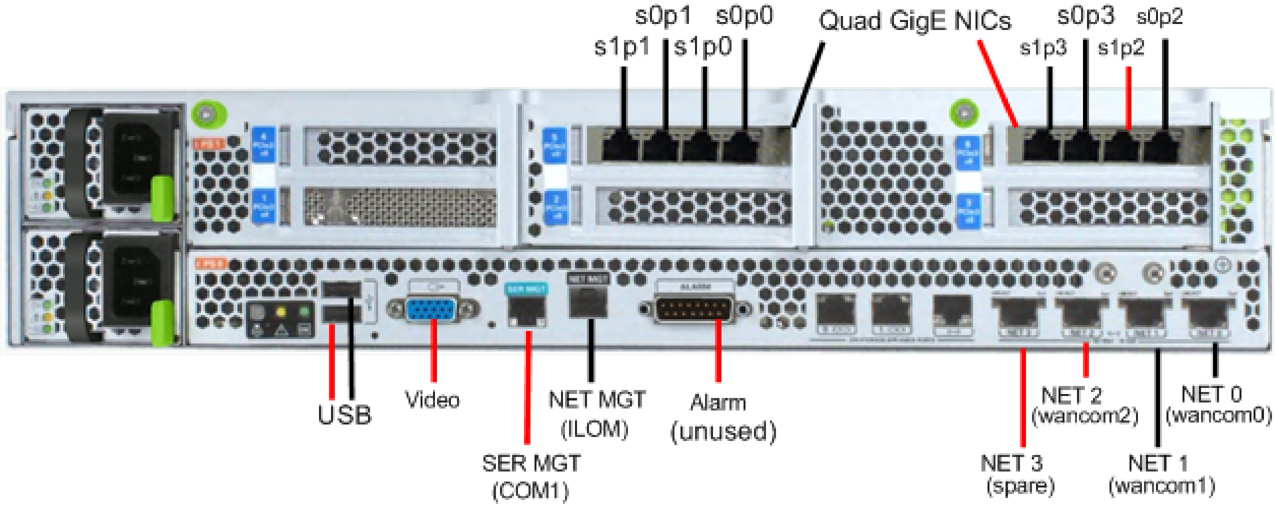
- Configuration A: Four onboard 10 GigE ports and no Quad GigE NIC
- Configuration B: Four onboard 10 GigE ports and 1 Quad GigE NIC
- Configuration C: Four onboard 10 GigE ports and 2 Quad GigE NICs



Netra Server X3-2 for Acme Packet Configuration A (4 Onboard 10 GigE Ports)



Netra Server X3-2 for Acme Packet Configuration B (4 Onboard 10 GigE Ports & 1 Quad GigE NIC)



Netra Server X3-2 for Acme Packet Configuration C (4 Onboard 10 GigE Ports & 2 Quad GigE NICs)

Oracle recommends using Category 6 (or better) for all Ethernet connections.

You can install and remove Ethernet and 1000BASE-T cables while the Netra Server X3-2 for Acme Packet is operational. Not every port needs to be utilized for proper operation. However, when a cable is disconnected and the link is lost, an alarm is generated.

Available Connections

Please read all of the information for each of the available connections prior to cabling the Netra X5-2 for Communications.

Port	Description	You Need:
NET (0-3)	10 GigE ports - labeled Net 3, Net 2, Net 1 and Net 0 (left to right). Enables you to connect the Netra X5-2 to your network.	A Category 6 (or better) Ethernet cable to connect to the NET 0 port to your network Network parameters such as an IP address (can be provided by DHCP services or assigned a static address in the OS) Additional Category 6 (or better) Ethernet cables and Ethernet addresses as needed for additional connections to NET 1 - 3
NET MGT	Provides a 10/100BASE-T Ethernet connection to the Service Processor (SP) through an RJ-45 connector. The NET MGT port provides support connections to the SP using the Oracle Integrated Lights Out Manager (ILOM) CLI and Web interface. By default, the NET MGT port is configured to use DHCP to automatically obtain an IP address. Alternatively, you can assign a static IP address to the NET MGT port. To use the NET MGT port, you must configure its network settings. Once configured, use the NET MGT port IP address to log on to the device using a browser or secure shell.	Category 6 (or better) Ethernet cable to connect the NET MGT port to your network IP address for this port (required from DHCP or a static address)
SER MGT (COM1)	Provides a TIA/EIA-232 serial Oracle/Cisco standard connection to the SP through an RJ-45 connector. Default settings: 8N1: eight data bits, no parity, one stop bit 115200 baud Disable hardware flow control (CTS/RTS) Disable software flow control (XON/XOFF)	A terminal device (e.g., terminal, connection to a terminal server, or computer such as a laptop running terminal emulation software) A cable to connect the terminal device to the SER MGT (COM1) port
USB	Provides USB connections to the SP. The USB ports are hot pluggable, so you can connect and disconnect USB cables from these ports and peripheral devices without affecting server operations.	USB keyboard USB mouse Note: Maximum USB cable length: 5 meters
VIDEO	Provides a temporary video connection to the SP.	VGA monitor HDB-15 video cable with a maximum cable length of 6 meters (19.7 feet)

Local Console Cabling Procedure

This section explains how to physically make a console connection to the Netra Server X3-2 for Acme Packet. Administration console may be connected to either the ILOM (NET MGT), the local VGA+USB console ports, or the local SER MGT (COM1) serial console port. When configuring bootloader parameters, set the console to VGA if ILOM or VGA+USB are used, or COM1 if SER MGT is used. The bootloader is accessible on all console ports, but only input from the active console port can be recognized by the Netra Server X3-2 for Acme Packet.

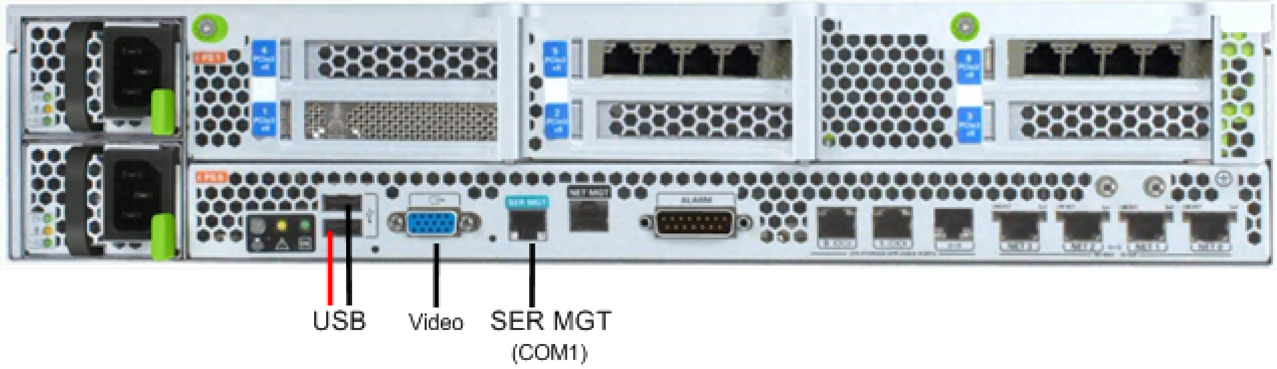
Note: DO NOT configure COM2 in the bootparams menu.

- To cable a serial console connection:
  - Serial console cable with an RJ-45 connector
- To cable a USB and Video Connection:
  - DB-15 video cable with a maximum cable length of 6 meters (19.7 feet)
  - USB cable with a maximum cable length of 6 meters (19.7 feet)
  - USB keyboard

In the following procedure, you have the option to either cable a serial connection or to cable a USB/Video connection.

To cable a local console connection:

1. Locate the appropriate cable(s) to connect to the Netra Server X3-2 for Acme Packet.
2. To cable a serial connection, insert the serial console cable into the SER MGT (COM1) port.



Connecting to USB, VGA and SER MGT (COM1) Ports

Note: Refer to the Netra Server X3-2 hardware documentation for information on how to configure your terminal application to connect to the console, and how to establish communications with the Netra Server X3-2 for Acme Packet.

3. To cable a USB/Video connection, insert the 15-pin connector on the end of the video cable into the Video port. Then insert the USB cable from the mouse and keyboard into the USB ports.
4. Lead the cables neatly away from the rear panel.
5. Plug in the cables to their respective destination components.

ILOM Cabling Procedure

This section explains how to make a connection to the Netra Server X3-2 for Acme Packet ILOM port. For a remote permanent connection to the SP over the ILOM connection, use the rear panel NET MGT port.

Refer to the Netra Server X3-2 for Acme Packet hardware documentation for information on how to configure your Web browser application to connect to the console, and how to establish communications with the Netra Server X3-2 for Acme Packet.

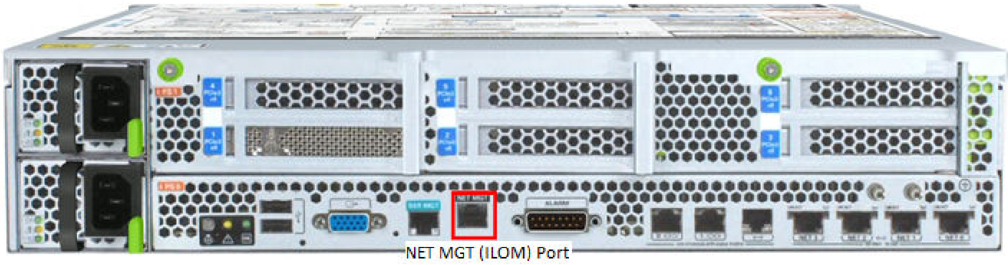
Note: Keep Ethernet cables separated from power cables by at least 60mm where possible and never run them in the same channel of the rack without segregation.

Prerequisites:

- Category 6 (or better) Ethernet cable with RJ-45 jacks

To cable an ILOM connection:

1. Locate the cable to connect to the Netra Server X3-2 for Acme Packet.
2. Plug the RJ-45 connector into the ILOM port.



3. Lead the cable neatly away from the rear panel.
4. Connect the other end of the cable to the LAN.

Network Management Ports Cabling Procedure

This section describes how to connect cables to the network management ports. These ports support 10/100/1G/10G Mbps speeds.

Note: Keep Ethernet cables separated from power cables by at least 60mm where possible and never run them in the same channel of the rack without segregation.

Prerequisites:

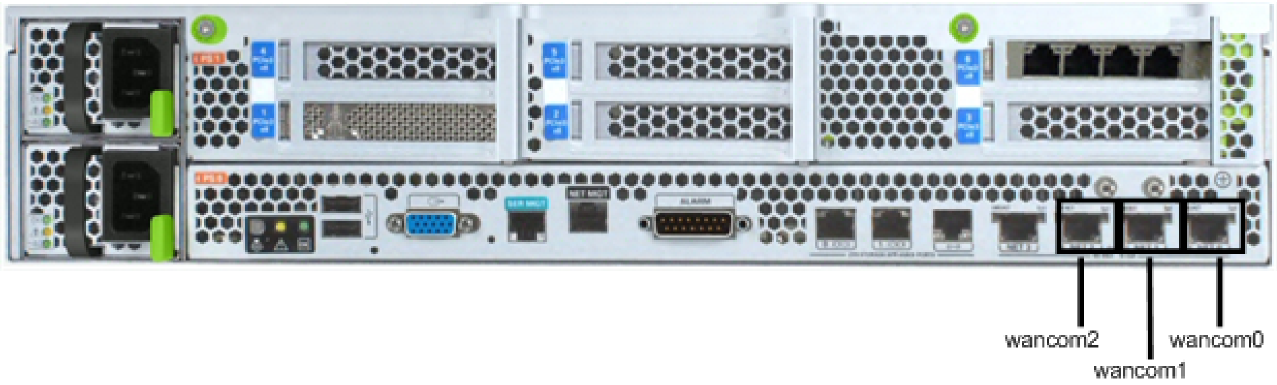
- Category 6 (or better) Ethernet cable with RJ-45 jacks

To connect to the network management ports:

1. Locate the Ethernet cables you plan to connect to the Netra Server X3-2 for Acme Packet.
2. Insert the RJ-45 connector on the end of the Ethernet cable into one the NET0 Ethernet port (`wancom0`).

   Note: The wancom0 and wancom1 ports are common to all supported Netra Server X3-2 for Acme Packet configurations. The wancom2 port is not used on the Oracle ECB.

   The release tab on the RJ-45 jack will click into place when you insert it properly.



Network Management Ports

3. Route the cable away from the Netra Server X3-2 for Acme Packet, ensuring that the Ethernet cables are not stretched tightly or subjected to extreme stress.

Media and Signaling Network Interfaces

This section explains how to cable the media and signaling ports. These ports accept copper GigE connectors.

Note: Perform all cabling procedures according to the established standards for your organization.

Category 6 (or better) Ethernet cables with RJ-45 jacks are used for connecting to the Netra Server X3-2 for Acme Packet media and signaling ports to your production network.

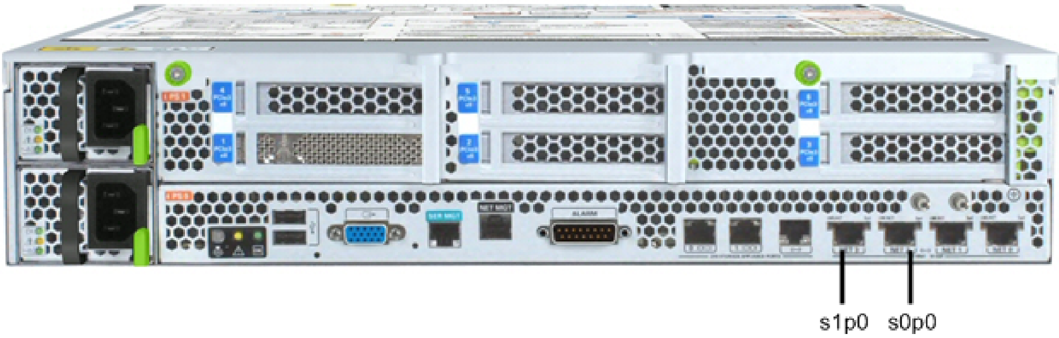
Regardless of configuration, media ports support 10/100/1000BASE-T only. Do not attempt to connect 10GBASE-T equipment to the signaling and media ports.

Prerequisites:

- Category 6 (or better) Ethernet cables with RJ-45 jacks

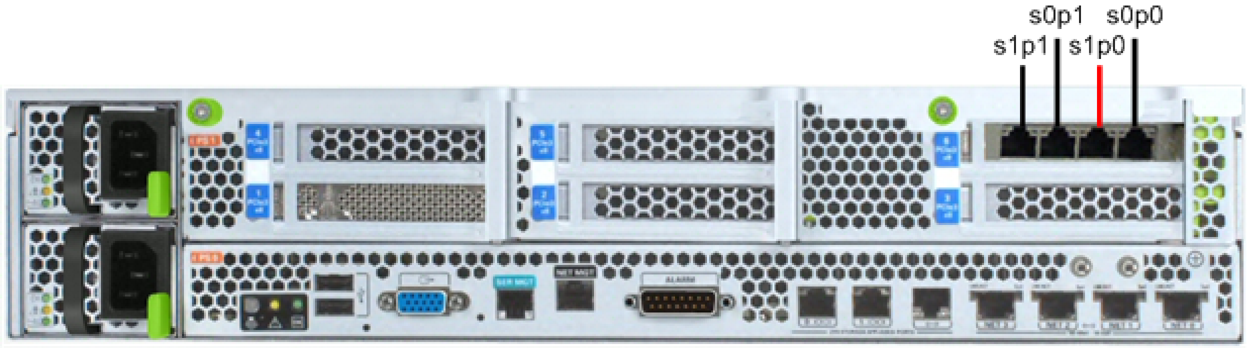
To connect to the media and signaling ports:

1. Locate the Ethernet cables you plan to connect to the media and signaling ports of the Netra Server X3-2 for Acme Packet.
2. Insert the RJ-45 connector on the end of the Ethernet cable into one of the 1000BASE-T copper media and signaling ports. The available signaling and media ports depend on the chosen configuration:
   - For configurations with no Quad GigE NICs, two onboard Ethernet ports are available for use as signaling and media ports as shown in the following.



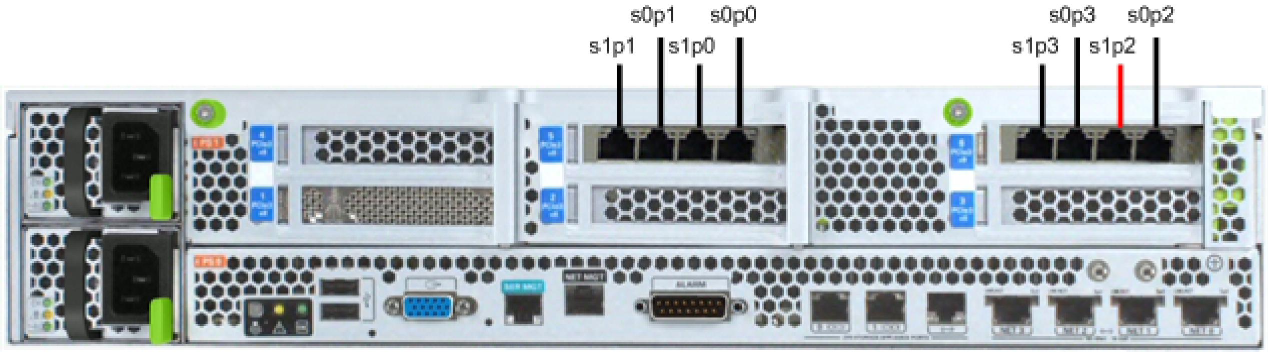
Supported Signaling/Media Ports (4 10 GigE Ports)

   - When the configuration consists of four onboard 10 GigE ports and one Quad GigE NIC, the signaling and media ports include **s1p1, s0p1, s1p0,** and **s0p0** as shown in the following. The release tab on the RJ-45 jack will click into place when you insert it properly.



Supported Signaling/Media Ports (4 OB 10 GigE Ports & 1 Quad GigE NIC)

   - When the configuration consists of four onboard 10 GigE ports and two Quad GigE NICs, the signaling and media ports include **s1p1, s0p1, s1p0, s0p0, s1p3, s0p3, s1p2** and **s0p2** as shown in the following. The release tab on the RJ-45 jack will click into place when you insert it properly.



Supported Signaling/Media Ports (4 OB 10 GigE Ports & 2 Quad GigE NICs)

3. Route the cable away from the Netra Server X3-2 for Acme Packet. Make sure that the Ethernet cables are not stretched tightly or subjected to extreme stress.
4. Repeat Steps 1 through 2 for each additional Ethernet cable you connect to your Netra Server X3-2 for Acme Packet.

Cabling for HA Deployments

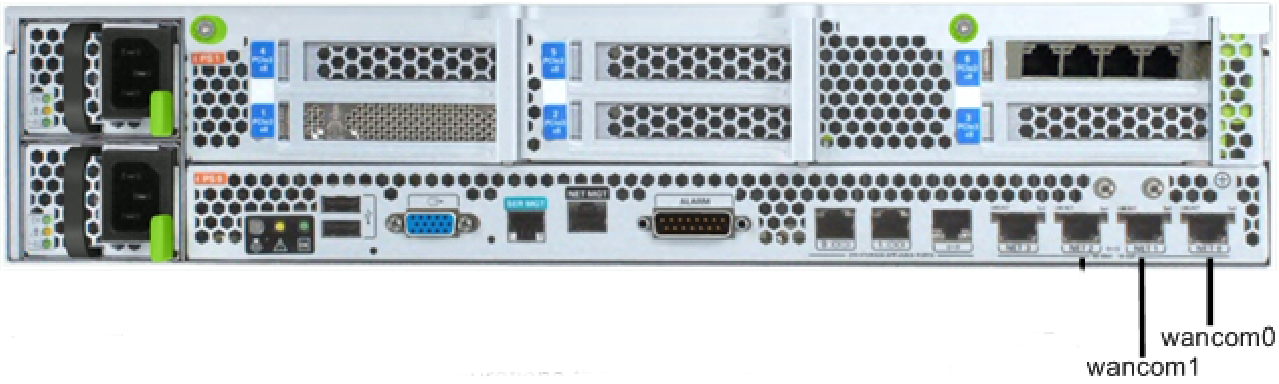
The information and instructions in this section explain how to cable a high availability (HA) node.

HA Cabling

Category 6 Ethernet cables are required for cabling two HA nodes together.

Rear Panel Cabling

You can use one connection for HA redundancy support between the two members of an HA node. As a rule, **wancom0** should be reserved as the boot/maintenance interface. This leaves **wancom1** available for sharing HA information.



4 Onboard 10 GigE Ports & 1 Quad GigE NIC

Prerequisites:

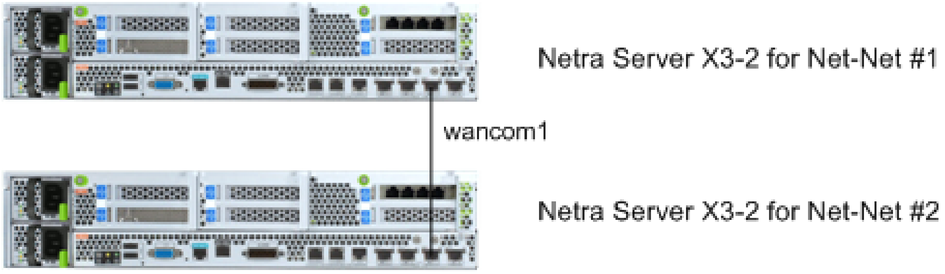
- Category 6 (or better) Ethernet cables with RJ-45 jacks

Single Rear Interface Support

To cable a Netra Server X3-2 for Acme Packet HA node using single rear interface support:

1. Insert one end of an Ethernet cable into **wancom1** on the rear panel of Netra Server X3-2 for Acme Packet #1. The release tab on the RJ-45 jack clicks into place when you insert it properly.

2. Insert the other end of the Ethernet cable into the corresponding management interface on the rear panel of the Netra Server X3-2 for Acme Packet #2 as presented here. For example, If you use **`wancom1`** on Netra Server X3-2 for Acme Packet #1, then you will connect it to **`wancom1`** on Netra Server X3-2 for Acme Packet #2.


Netra Server X3-2 for Net-Net #1
wancom1
Netra Server X3-2 for Net-Net #2

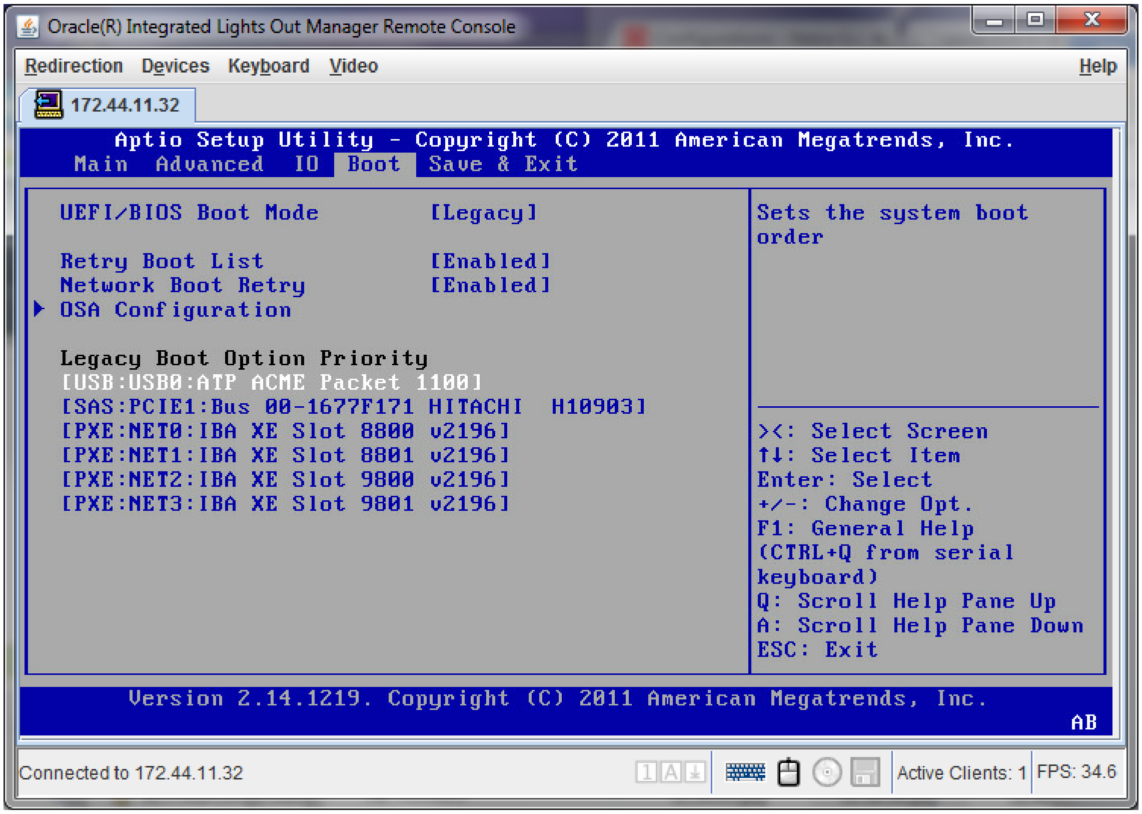
HA Node Using Single Rear Interface Support (No Quad GigE NIC)

3. Refer to the configuration procedures located in the HA Nodes information in this Configuration Guide.

BIOS Setting Configuration

The following changes on the Netra Server X3-2 are required to run Oracle Enterprise Communications Broker. This procedure shows where to make changes in the BIOS setup utility.

1. Set the USB slot to be the first boot device. The disk controller should be the second boot device.


Oracle(R) Integrated Lights Out Manager Remote Console
Redirection
Devices
Keyboard
Video
Help
172.44.11.32
Connected to 172.44.11.32
Active Clients: 1
FPS: 34.6

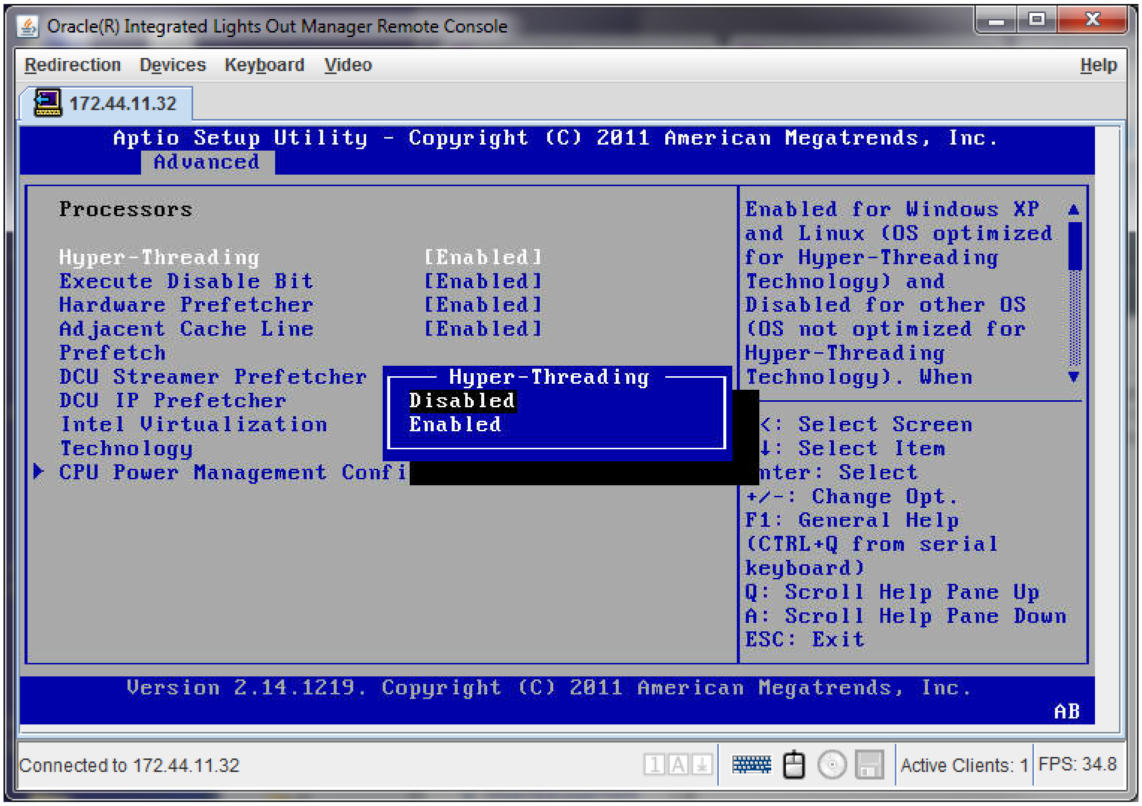
```
        Aptio Setup Utility - Copyright (C) 2011 American Megatrends, Inc.
    Main  Advanced  IO  Boot  Save & Exit

  UEFI/BIOS Boot Mode          [Legacy]         Sets the system boot
                                                order
  Retry Boot List              [Enabled]
  Network Boot Retry           [Enabled]
▶ OSA Configuration

  Legacy Boot Option Priority
  [USB:USB0:ATP ACME Packet 1100]
  [SAS:PCIE1:Bus 00-1677F171 HITACHI  H10903]
  [PXE:NET0:IBA XE Slot 8800 v2196]             ><: Select Screen
  [PXE:NET1:IBA XE Slot 8801 v2196]             ↑↓: Select Item
  [PXE:NET2:IBA XE Slot 9800 v2196]             Enter: Select
  [PXE:NET3:IBA XE Slot 9801 v2196]             +/-: Change Opt.
                                                F1: General Help
                                                (CTRL+Q from serial
                                                keyboard)
                                                Q: Scroll Help Pane Up
                                                A: Scroll Help Pane Down
                                                ESC: Exit

        Version 2.14.1219. Copyright (C) 2011 American Megatrends, Inc.
                                                                     AB
```

2. Set Hyper-Threading to **`Disabled`**

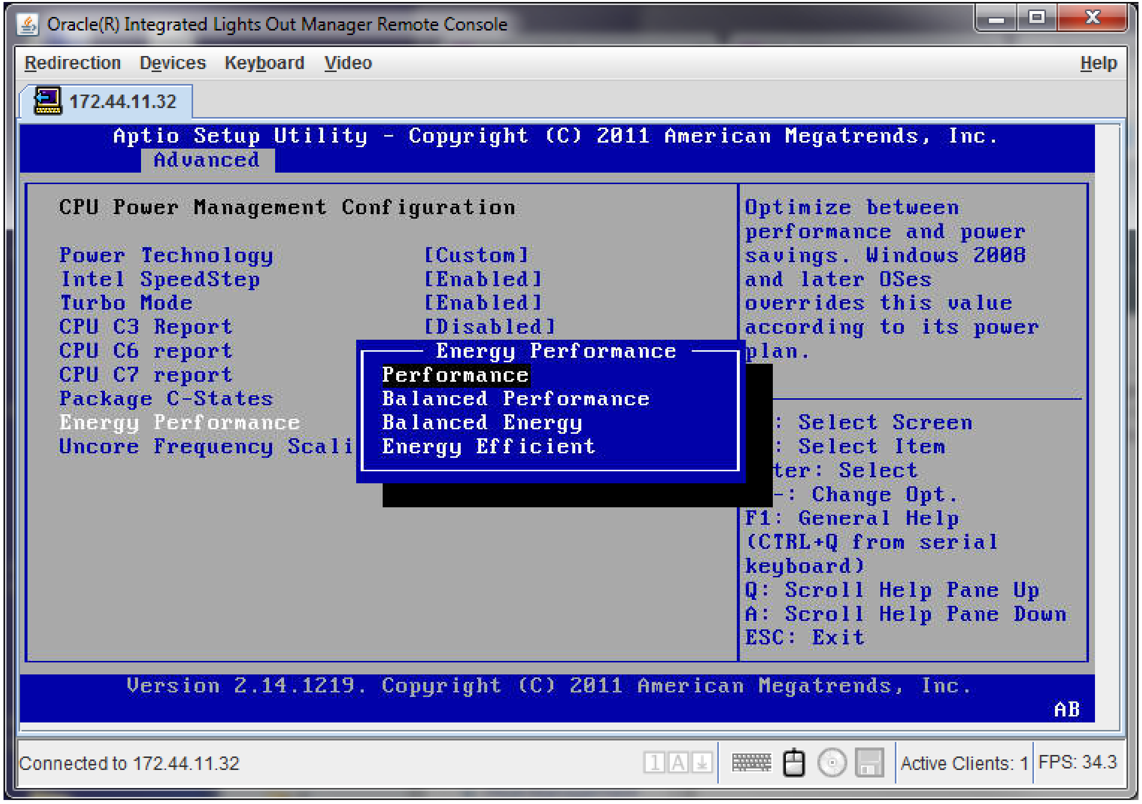

Oracle(R) Integrated Lights Out Manager Remote Console
Redirection
Devices
Keyboard
Video
Help
172.44.11.32
Connected to 172.44.11.32
Active Clients: 1
FPS: 34.8

```
        Aptio Setup Utility - Copyright (C) 2011 American Megatrends, Inc.
              Advanced

  Processors                                    Enabled for Windows XP
                                                and Linux (OS optimized
  Hyper-Threading              [Enabled]        for Hyper-Threading
  Execute Disable Bit          [Enabled]        Technology) and
  Hardware Prefetcher          [Enabled]        Disabled for other OS
  Adjacent Cache Line          [Enabled]        (OS not optimized for
  Prefetch                                      Hyper-Threading
  DCU Streamer Prefetcher  ── Hyper-Threading ─ Technology). When
  DCU IP Prefetcher         Disabled
  Intel Virtualization      Enabled             <: Select Screen
  Technology                                    ↓: Select Item
▶ CPU Power Management Confi                    nter: Select
                                                +/-: Change Opt.
                                                F1: General Help
                                                (CTRL+Q from serial
                                                keyboard)
                                                Q: Scroll Help Pane Up
                                                A: Scroll Help Pane Down
                                                ESC: Exit

         Version 2.14.1219. Copyright (C) 2011 American Megatrends, Inc.
                                                                      AB
```

3. Change Energy Performance to **Performance**


Oracle(R) Integrated Lights Out Manager Remote Console
Redirection
Devices
Keyboard
Video
Help
172.44.11.32
Connected to 172.44.11.32
Active Clients: 1
FPS: 34.3

```
        Aptio Setup Utility - Copyright (C) 2011 American Megatrends, Inc.
              Advanced

  CPU Power Management Configuration            Optimize between
                                                performance and power
  Power Technology             [Custom]         savings. Windows 2008
  Intel SpeedStep              [Enabled]        and later OSes
  Turbo Mode                   [Enabled]        overrides this value
  CPU C3 Report                [Disabled]       according to its power
  CPU C6 report          ── Energy Performance ─ plan.
  CPU C7 report           Performance
  Package C-States        Balanced Performance
  Energy Performance      Balanced Energy        : Select Screen
  Uncore Frequency Scali  Energy Efficient       : Select Item
                                                ter: Select
                                                -: Change Opt.
                                                F1: General Help
                                                (CTRL+Q from serial
                                                keyboard)
                                                Q: Scroll Help Pane Up
                                                A: Scroll Help Pane Down
                                                ESC: Exit

         Version 2.14.1219. Copyright (C) 2011 American Megatrends, Inc.
                                                                      AB
```

4. After setting Performance, press **Escape** to return to the main menu, then select **Save & Exit** to apply the changes.
   The system will reboot using the newly configured settings.

Virtual Systems

The Oracle Enterprise Communications Broker (ECB) Software Only distribution is designed to operate on virtual machines running on generic, off-the-shelf servers. Oracle recommends using the Oracle Virtual Machine (OVM) hypervisor for running the ECB virtual application.

You can install the virtual machine software on the hardware of your choice. The number of VMs supported by a server is constrained only by the resources on your system.

Minimum VM Resources

Each VM instance requires the following minimum allocation or network resources.

- CPU cores: 2
- Memory: 8GB RAM
- Hard drive storage: 40GB
- Interfaces: 4 recommended

Format Hard Drive

Run the command **`format-hard-drive`**, as described in the *Oracle® Enterprise Session Director ACLI Configuration Guide* immediately after successful installation.

2

Getting the System Operational

Appliance Installation and Start-Up

This section outlines hardware installation at a very high level and describes system power-on. It bridges hardware installation and application start-up, presenting information about what to expect from Oracle Enterprise Communications Broker software as the hardware powers up. Administrators need to know how to access the software while it boots, and what successful software startup looks like.

If running the Oracle Enterprise Communications Broker as a virtual application, refer to the hardware vendor's installation instructions for hardware to learn how to access the software while it boots. From a console connection, there is little difference to the way successful startup appears as an appliance versus a virtual machine.

Hardware Installation Summary

Installing your Oracle Enterprise Communications Broker in your rack requires the steps summarized here. This checklist is only an overview. It is not designed to substitute for following the detailed procedures in the hardware installation guides.

1. Unpacking the Oracle Enterprise Communications Broker
2. Installing the Oracle Enterprise Communications Broker into your rack
3. Installing power supplies
4. Installing fan modules
5. Installing physical interface cards
6. Cabling the Oracle Enterprise Communications Broker

   Make sure you complete installation procedures fully and note the safety warnings to prevent physical harm to yourself and/or damage to your Oracle Enterprise Communications Broker.

   After you have completed the hardware installation procedures, you are ready to establish a connection to your Oracle Enterprise Communications Broker. Then you can load the Oracle Enterprise Communications Broker software image you want to use and establish basic operating parameters.

Connecting to The Oracle Enterprise Communications Broker

By default, Oracle delivers the Oracle Enterprise Communications Broker (ECB) with no management IP address. You must set this address the first time you start the system. See the System Boot section.

You can connect to the ECB through a direct console connection or by creating a remote SSH session. Both methods provide a wide range of configuration, monitoring, and management options. IP-based management access, including SSH and the web GUI, requires an IP address for your management port. This address is specified in the **ip address** boot parameter.

Note: The system displays the **ip address** parameter with different names, depending on the context:

- The boot parameters wizard field name is also **ip address**.
- The initial configuration wizard field name is **Management interface ip address**.
- The ACLI **show interfaces** command field name is **wancom0**.

By default, SSH, SFTP, and web GUI connections to the ECB are enabled, but are only accessible by way of the **ip address**. You cannot use SSH, SFTP, or the web GUI until you set this address.

Depending on the platform, you may need to install the software installation upon first startup. You perform and monitor software installation by way of the console connection. The ECB requires most configuration by way of the web GUI. Procedures requiring the ACLI include:

- Change default management interface IP address
- Format hard drive
- Set and change password
- Set and change SIP Monitor and Trace filters

Local Connections and Time-outs

The ACLI is available through serial and SSH connections. Prior to software installation, you reach the ACLI through a local, serial connection.

When deploying the Oracle Enterprise Communications Broker (ECB) on a virtual machine, the virtual machine manager provides console access through a virtual serial connection. See documentation on your virtual machine to learn how to access the console. Working with the virtual machine console is the same as working on dedicated hardware.

When deploying on dedicated hardware, refer to the hardware documentation "Applicable Platforms" for instructions on connecting to the ECB console.

Plug one end of the cable into your terminal and the other end into the RJ-45 port, normally located on the back of your server.

To set up a console connection to the ECB:

1. Set the connection parameters for your terminal to the default boot settings:
    - Baud rate: 115,200 bits/second
    - Data bits: 8
    - Parity: No
    - Stop bit: 1
    - Flow control: None
2. Use a serial cable to connect your PC to the ECB. Refer to your hardware documentation for the location of your server's serial port.
3. Power on the ECB.
   The system boots. Upon successful boot, the system prompts you to log on.

   ```
   Password:
   ```
4. Enter the appropriate password information when prompted to log into User mode of the ACLI. The default user mode password is **`acme`**.

   The system displays the ACLI's user mode prompt :

   ```
   ORACLE>
   ```
5. If necessary, enter Superuser mode by entering **`enable`** at the ACLI and pressing Enter.

The system ACLI prompts you for the superuser password:

```
ORACLE>enable
Password:
```

6. Enter the appropriate password information to log into Superuser mode of the ACLI. The default Superuser mode password is **`packet`**.
   The system changes the ACLI prompt to:

```
ORACLE#
```

7. Proceed with system configuration or setup.

You can control the amount of time it takes for your console connection to time out by setting the **`console-timeout`** parameter in the system configuration. When your connection times out, the ECB displays the login sequence again and prompts you for your passwords. The default for this field is 0, which means that no time-out is being enforced.

SSH Connections and Time-outs

You can use SSH to connect to the Oracle Enterprise Communications Broker (ECB) and provision the ECB remotely through the management interface over IP. You configure the management interface IP during system setup, or by way of the ECB boot parameters.

The Oracle Enterprise Communications Broker can support up to five concurrent SSH and SFTP sessions. Note that only one user can carry out configuration tasks at a time.

To connect to the ECB, you need to know the IP address of its administrative interface (wancom0). You can find the ECB wancom0 IP address by using the ACLI to display the boot parameter value named **IP Address**.

You can manage the SSH connections to the ECB by setting certain ACLI parameters and by using certain commands:

- To view the users who are currently logged into the system, use the **`show users`** command. You can see the ID, timestamp, connection source, and privilege level for active connections.
- From Superuser mode in the ACLI, you can terminate the connections of other users to free up connections. Use the **`kill user`** command, with the corresponding connection ID.
- When you reboot the ECB from an SSH session, you lose IP access and the connection.

Initiate SSH without Username and Password

Many SSH clients allow you to initiate an SSH connection without specifying a username. To initiate an SSH connection to the Oracle Enterprise Communications Broker (ECB) without specifying usernames and SSH user passwords:

1. Open your SSH client.
2. At the prompt in the SSH client, type the **`ssh`** command, a space, the IPv4 address of your Oracle Enterprise Communications Broker, and press Enter.

   The SSH client prompts you for a password before connecting to the ECB. Enter the ECB User mode password. After authentication, an SSH session is initiated and you can continue with tasks in User mode or enable Superuser mode.

   Note that some clients interpret SSH session initiation without a Username as a means of logging in with your system login name. The preceding procedure does not work for such clients.

   Note: You can also create connections to the ECB using additional Username and password options.

SSH with Username and Password

To initiate an SSH connection to the Oracle Enterprise Communications Broker with an SSH username and password:

1. In the ACLI at the Superuser prompt, type the **ssh-password** and press Enter. Enter the name of the user you want to establish. Then enter a password for that user when prompted. Passwords do not appear on your screen.

   ```
   SYSTEM# ssh-password
   SSH username [saved]: MJones
   Enter new password: 95X-SD
   Enter new password again: 95X-SD
   ```

   Note: After you configure ssh-password, the SSH login accepts the username and password you set, as well as the default SSH/SFTP usernames: User and admin.

2. Configure your SSH client to connect to your Oracle Enterprise Communications Broker's management IPv4 address using the username you just created. The standard version of this command would be:

   ```
   ssh -l MJones 10.0.1.57
   ```

3. Enter the SSH password you set in the ACLI.

   ```
   MJones@10.0.2.54 password: 95X-SD
   ```

4. Enter your User password to work in User mode on the Oracle Enterprise Communications Broker. Enable Superuser mode and enter your password to work in Superuser mode.
5. An SSH session window opens and you can enter your password to use the ACLI.

GUI Access

To access theOracle Enterprise Communications Broker (ECB) for ongoing configuration and management, you must use the GUI. The system allows only a few user and provisioning procedures by way of the ACLI, such as setting the initial management IP address and changing GUI access passwords. The system does not allow disabling the GUI.

You can configure GUI access by way of HTTP or HTTPS at the configured management address, which you must set prior to attempting to log on.

When a user accesses the GUI, the ECB displays the log on screen. Upon successful log on, the system allows access to the System Administration and Service Provisioning controls.

Setting Your Login Banner

The Oracle Enterprise Communications Broker allows the user to create and edit the message displayed in the Login banner dialog, which appears upon successful login.

1. Click the **Configuration** tab.
   The Oracle Enterprise Communications Broker displays the configuration panel.
2. Click the **Wizards** dropdown.
   The Oracle Enterprise Communications Broker displays the widget menu panel.
3. Click the **Set login banner** link.
   The Oracle Enterprise Communications Broker displays the **Set login banner** dialog, which includes a text box allowing the user to write a login message.
4. Type your banner text and click the **Save** button to set the banner.
   The Oracle Enterprise Communications Broker sets the login banner.

System Boot

Whenever your Oracle Enterprise Communications Broker boots, the following information about the tasks and settings for the system appear in your terminal window.

- System boot parameters
- From what location the software image is being loaded: an external device or internal flash memory
- Requisite tasks that the system is starting
- Log information: established levels and where logs are being sent
- Any errors that might occur during the loading process

After the loading process is complete, the ACLI login prompt appears.

Note: You can set boot parameters using the ACLI or the GUI. Boot parameter definitions, which help you understand what you should set them to, are provided below.

Oracle Enterprise Communications Broker Boot Parameters

Boot parameters specify the information that your Oracle Enterprise Communications Broker uses at boot time when it prepares to run applications. The Oracle Enterprise Communications Broker's boot parameters:

- Allow you to set the IP address for the management interface (wancom0).
- Allow you to set a system prompt. The target name parameter also specifies the title name displayed in your web browser and SNMP device name parameters.
- Determine the software image to boot and from where the system boots that image.
- Sets up the username and password for network booting from an external FTP server.

In addition to providing details about the Oracle Enterprise Communications Broker's boot parameters, this section explains how to view, edit, and implement them.

When displaying the boot parameters, your screen shows a help menu and the first boot parameter (boot device). Press Enter to continue down the list of boot parameters.

Boot Parameter Changes

You can access and edit boot parameters by using either the ACLI or by interrupting the system boot process.

Note: Changes to boot parameters do not go into effect until you reboot the Oracle Enterprise Communications Broker.

Oracle recommends that you use management port 0 (wancom0) as the boot interface, and that your management network is either:

- directly a part of your LAN for management port 0
- accessible through management port 0

Otherwise, your management messages may use an incorrect source address.

Change Boot Parameters from the ACLI

To access and change boot parameters from the ACLI:

1. In Superuser mode, type configure terminal, and press Enter.

   ```
   ORACLE# configure terminal
   ```

2. Type bootparam, and press Enter. The boot device parameters display.

   ```
   ORACLE(configure)# bootparam
   '.' = clear field;  '-' = go to previous field;  ^D = quit
   Boot File         : /boot/nnPCz100.gz
   ```

   To navigate through the boot parameters, press Enter and the next parameter appears on the following line.

   You can navigate through the entire list this way. To go back to a previous line, type a hyphen (-) and press Enter. Any value that you enter entirely overwrites the existing value and does not append to it.

3. To change a boot parameter, type the new value that you want to use next to the old value. For example, if you want to change the image you are using, type the new filename next to the old one. You can clear the contents of a parameter by typing a period and then pressing Enter.

   ```
   ORACLE(configure)# bootparam
   '.' = clear field;  '-' = go to previous field;  ^D = quit
   Boot File         : /boot/nnPCz100.gz /boot/nnPCz200.gz
   ```

When you have scrolled through all of the boot parameters, the system prompt for the configure terminal branch displays.

```
ORACLE(configure)#
```

4. Exit the configure terminal branch.
5. Reboot the Oracle Enterprise Communications Broker for the changes to take effect.

The ACLI **reboot** and **reboot force** commands initiate a reboot. With the **reboot** command, you must confirm that you want to reboot. With the **reboot force** command, you do not have make this confirmation.

```
ORACLE# reboot force
```

The Oracle Enterprise Communications Broker completes the full booting sequence. If necessary, you can stop the auto-boot at countdown to fix any boot parameters.

If you configured boot parameters correctly, the system prompt displays and you can go ahead with configuration, management, or monitoring tasks.

Note: If you configured the boot parameters incorrectly, the Oracle Enterprise Communications Broker goes into a booting loop and displays an error message.

```
Error loading file: errno = 0x226.
Can't load boot file!!
```

Press the space bar to stop the loop. Correct the error in the boot parameter, and reboot the system.

Change Boot Parameters by Interrupting a Boot in Progress

To access and change boot parameters by interrupting a boot in progress:

1. When the Oracle Enterprise Communications Broker is in the process of booting, you can press the space bar on your keyboard to interrupt when you see the following message appear:

```
Press the space bar to stop auto-boot...
```

2. After you stop the booting process, you can enter the letter p to display the current parameters, the letter c to change the boot parameters or the @ (at-sign) to continue booting.

```
[Acme Packet Boot]: c
'.' = clear field;  '-' = go to previous field;  ^D = quit
Boot File        : /boot/bzImage-bones64
```

To navigate through the boot parameters, press Enter and the next parameter appears on the following line.

You can navigate through the entire list this way. To go back to a previous line, type a hyphen (-) and press Enter. Any value that you enter entirely overwrites the existing value and does not append to it.

3. To change a boot parameter, type the new value that you want to use next to the old value. For example, if you want to change the image you are using, type the new filename next to the old one.

```
[Acme Packet Boot]: c
'.' = clear field;  '-' = go to previous field;  ^D = quit
Boot File        : /boot/bzImage-bones64 /boot/bzImage.gz
```

4. After you have scrolled through the complete list of boot parameters, you return to the boot prompt. To reboot with your changes taking effect, type @ (the at-sign), and press Enter.

```
[Acme Packet Boot]: @
```

The Oracle Enterprise Communications Broker completes the full booting sequence, unless there is an error in the boot parameters.

If you have configured boot parameters correctly, the system prompt displays and you can go ahead with configuration, management, or monitoring tasks.

Note: If you have configured the boot parameters incorrectly, the Oracle Enterprise Communications Broker goes into a booting loop and displays an error message.

```
Error loading file: errno = 0x226.
Can't load boot file!!
```

Press the space bar to stop the loop. Correct the error, and reboot your system.

Set Management IP Address

You must manually set your management IP address within the Oracle Enterprise Communications Broker's boot parameters.

To set your management interface IP, access the boot parameters using a serial console connection within the context of one of the methods described above.

1. Type the letter c (change) to start boot parameter editing.
2. Press Enter until you reach the parameter named **IP Address**.
3. Type in the desired IP address.
4. Press Enter until you reach the end of the boot parameter list.
5. Reboot your Oracle Enterprise Communications Broker.

After being set, the management interface IP address provides access to your system via ssh and the web GUI. You can verify the status of this interface using the following command to display the address and status of wancom0.

```
Oracle ECB# show interfaces brief
Slt Prt Vlan Interface  IP                     Gateway             Adm  Oper
Num Num   ID Name       Address                Address             Stat Stat
--- --- ---- ---------- ---------------------- ------------------- ---- ----
-   -    -   lo         127.0.0.1              -                    up   up
-   -    -   wancom0    122.30.204.127/16      -                    up   up
0   0    0   M00        122.170.1.200/16       0.0.0.0              up   up
----------------------------------------------------------------------------
Oracle ECB#
```

Format Hard Drive

Manual software installation, performed on virtual and COTs machines, does not include formatting the hard drive automatically. After manual software installation and boot parameter configuration, the user must format the hard drive from the ACLI.

Generic installation documentation may not include the requirement to format the hard-disk. Run the command **format hard-disk** from the Oracle Enterprise Communications Broker ACLI to create a persistent partition for your /opt directory, within which you can store data needed after a reboot. Perform this procedure the FIRST time you start your Oracle Enterprise Communications Broker.

Partial output is presented below. Be sure to accept all defaults presented during the format by typing the letter **y** when prompted.

```
ORACLE# format hard-disk
WARNING: Please ensure device is not currently in use by any applications
before proceeding
Continue [y/n]?: y
The following system partitions will now be created:
1: /opt 8000000 bytes
2: /crash 16218284032 bytes
Create the system partitions and filesystems as configured above [y/n]?: y
```

After the drive(s) are formatted, the system mounts the newly created partitions.

System Image Filename

The system image filename is a name you set for the image. This is also the filename the bootloader uses whenever booting your system. This filename must match the filename specified in the boot parameters.

When your image is located on your Oracle Enterprise Communications Broker, the parameter should start with /boot/ to indicate that the Oracle Enterprise Communications Broker is booting from it's local /boot directory.

If the filename set in the boot parameters does not point to the image you want sent to the Oracle Enterprise Communications Broker via SFTP, then you could not only fail to load the appropriate image, but you could also load an image from a different directory or one that is obsolete for your purposes. This results in a boot loop condition that you can fix by stopping the countdown, entering the appropriate filename, and rebooting the Oracle Enterprise Communications Broker.

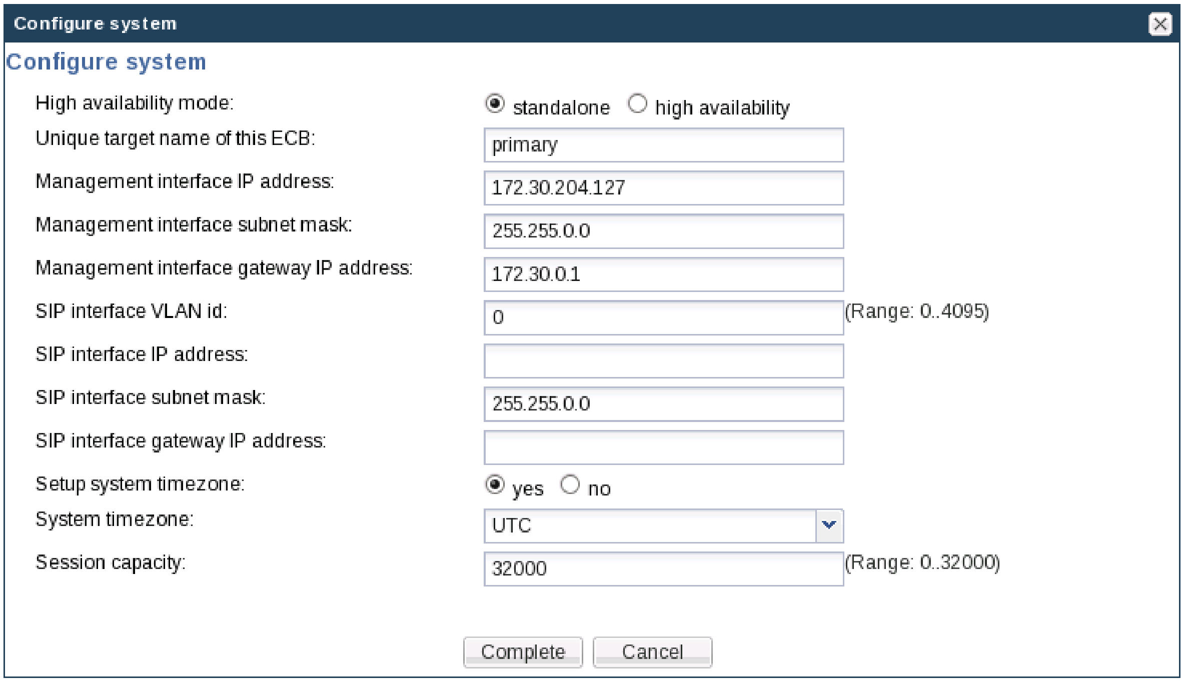
Initializing Your System

The Oracle Enterprise Communications Broker provides a means of initializing the system from the GUI. This procedure is required upon first startup. The user may need to initialize at other times, but must be aware that all configuration is lost (initialized) and that the system reboots.

With respect to getting a high availability configuration operational, note that you use the Set initial configuration wizard to configure the primary Oracle Enterprise Communications Broker first. Assuming the configuration is correct, high availability operations begin as soon as you Set initial configuration on the secondary Oracle Enterprise Communications Broker and that system completes its subsequent reboot.

1. To initialize your system, navigate to the Configuration screen and select **Set initial configuration** wizard from the wizard drop-down list.

   The system displays the Configure system dialog, allowing you to make all the settings needed to initialize it.



Configure system

Configure system

High availability mode:	● standalone ○ high availability
Unique target name of this ECB:	primary
Management interface IP address:	172.30.204.127
Management interface subnet mask:	255.255.0.0
Management interface gateway IP address:	172.30.0.1
SIP interface VLAN id:	0 (Range: 0..4095)
SIP interface IP address:	
SIP interface subnet mask:	255.255.0.0
SIP interface gateway IP address:	
Setup system timezone:	● yes ○ no
System timezone:	UTC
Session capacity:	32000 (Range: 0..32000)

   This wizard deletes any existing configuration and reboots the system when you click the **Complete** button.

2. **High availability mode**— Click the radio button that corresponds with your high availability configuration.

   - standalone—You have a single Oracle Enterprise Communications Broker.
   - high availability—You are deploying Oracle Enterprise Communications Brokers in pairs, connecting them together and configuring one as primary and the other as a secondary. The secondary can automatically take over for the primary, providing "hitless", redundant operation.

3. **Unique target name of this ECB**—Type the name of this system. This setting has an operational impact on your high availability configuration.
4. **Management interface IP address**—Enter the IP address to be used for accessing the Web GUI, and press Enter.
5. **Management interface subnet mask**—Enter subnet mask to be used for accessing the Web GUI, and press Enter.
6. **Management interface gateway IP address**—Enter the IP address to be used for reaching this network's gateway and press Enter.
7. **SIP interface VLAN id**—Enter the VLAN ID (0 to 4095), if any, required for operation on the network of your SIP interface.
8. **SIP interface IP address**—Enter the IP address to be used for accessing the SIP interface, and press Enter. This step is required; the system does not allow you to proceed without making a setting.
9. **SIP interface subnet mask**—Enter subnet mask to be used for accessing the SIP interface, and press Enter.
10. **SIP interface gateway IP address**—Enter the gateway IP address and press Enter.
11. **Setup system timezone**—Click the **yes** radio button to set the system timezone. Click **no** to skip this step.
12. **System timezone**—Select your timezone from the drop-down list.
13. **Session capacity**—Type in the number of sessions you purchased for this Oracle Enterprise Communications Broker.
14. Click the **Complete** button to proceed with deleting the existing configuration, setting the values in your wizard and rebooting your Oracle Enterprise Communications Broker. Click **Cancel** to cancel system initialization.

Adding a License with the Set License Wizard

TLS is the only software feature for which you need a license on the Oracle Enterprise Communications Broker. You must obtain a TLS license before you can add it. To obtain a license, you must present the correct system serial number to Oracle for your license to be generated.

1. From Configuration home, select **Set license** from the **Wizards** drop-down list.
   The Oracle Enterprise Communications Broker displays the **Set license** dialog.
2. Copy the serial number for your Oracle Enterprise Communications Broker and contact your customer support by logging into My Oracle Support or calling Oracle Customer support to make the request. Oracle replies shortly after with your license.
3. Having received your license from Oracle, enter your license in the Add license field.
   The system checks the license and, if correct, installs it. If the license is incorrect, the system tells you.

Setting Up System Basics

Before configuring and deploying your Oracle Enterprise Communications Broker, you might want to establish some basic attributes such as new User and Superuser passwords and system prompt.

New User and Superuser Passwords

ACLI passwords provide access for, SSH, SFTP and GUI sessions. Common security practices include changing these passwords from their defaults and at intervals defined by your organization. Refer to the ACLI's `secret` command, documented in the *Oracle Communications Session Border Controller ACLI Reference Guide* for information about changing user and superuser passwords.

New System Prompt

You can set the ACLI system prompt using **Configure system** or the **Set boot parameters** Wizard. Change the **`target name`** value to make it meaningful within your network. The target name may be up to 38 characters. A value that identifies the system in some way is often helpful.

3

Initial Configuration

Initial Configuration

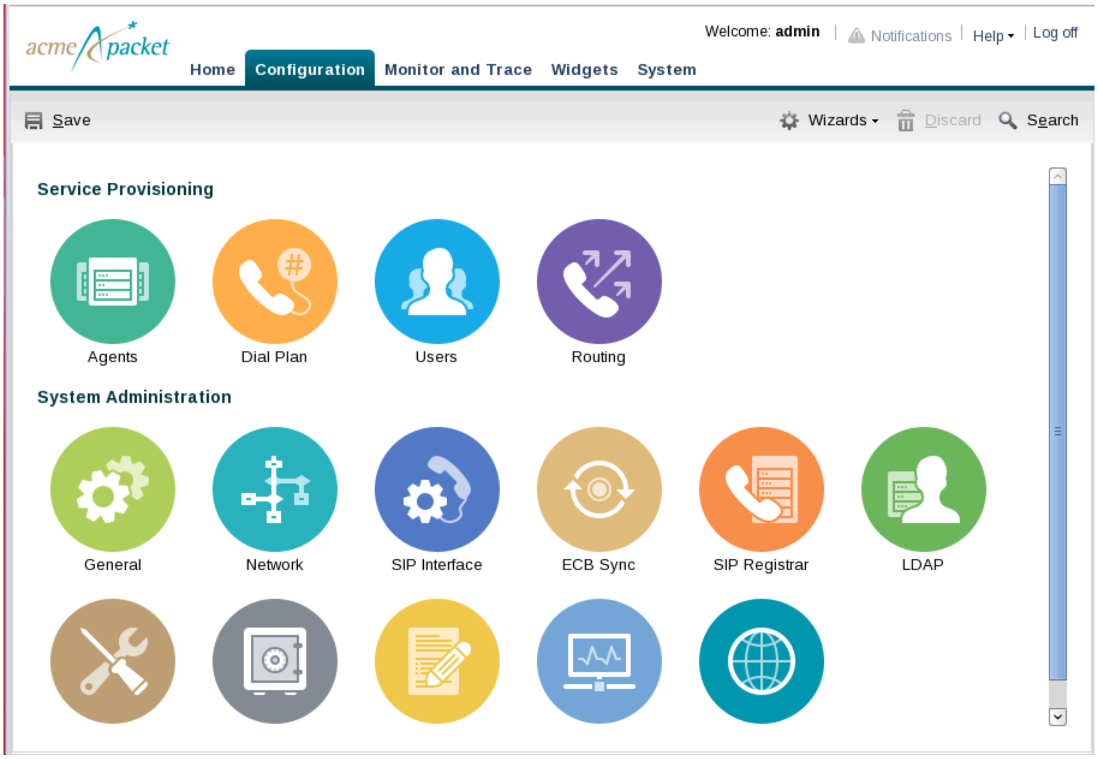
The initial configuration establishes system operations. You can perform the initial configuration of the Oracle Enterprise Communications Broker (ECB) from the GUI.

After you establish system operations, refer to the *Oracle Enterprise Communications Broker User's Guide* and the *Oracle Enterprise Communications Broker User's Guide* to configure SIP services and operations, as well as system file management and administrative functions.

System Administration

The Oracle Enterprise Communications Broker GUI collects controls for administering your system under System Administration, which are covered herein. In contrast, The Oracle Enterprise Communications Broker GUI collects tools used by network architects and service provisioning technicians under Service Provisioning. Service provisioning is the focus of the *Oracle Enterprise Communications Broker User's Guide*.

Regardless, all icons are explained below for context.



Configuration Icons

The following information provides high-level descriptions of the Service Provisioning and System Administration controls on theOracle Enterprise Communications Broker (ECB) Configuration tab.

Service Provisioning

The Service Provisioning icons provide access to the configuration objects used to provision service.

- Agents—Add agents that specify SIP and ENUM devices. An agent is usually a SIP-aware device that serves as a transit target or source for signaling managed by the Oracle Enterprise Communications Broker. Agents are often specified as next-hops for the purposes of routing.
- Dial plan—Add multiple dialing-contexts and dial-patterns. Dialing-contexts define the system behavior for calls placed to and from either a corporate of geographic focus. Dialing-contexts include multiple dial-patterns, which define the normalization required to most effectively manage diverse signaling structures.
- Users—Add user and other key phone numbers associated with the enterprise. The user database can specify each entry’s number or pattern, dialing context, agent, and policy, which can provide a starting point for processing the logic behind a user’s call treatment.
- Policy—Add policies that specify the codec and time conditions, as well as the routing, redirect, outbound translation, constraints, header normalization, and cnam masking actions.
- Routing—Add routing tables. Routing entries specify strict paths for signaling traffic, allowing you to specify policy and cost for traffic based on source and destination.

System Administration

The System Administration icons provide access the objects used to configure system operation.

- General—Specify standard system management information parameters, such as system identification information, system management information interfaces (SNMP and Syslog), and global service configurations including Denial of Service and High Availability settings.
- Network—Specify your network and High Availability settings, and add host routes.

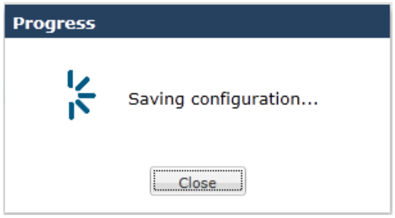
- SIP Interface—Specify the SIP interface and add SIP service ports. Configure SIP monitoring and SIP monitoring filters.
- ECB Sync—Specify Sync configuration settings and add Sync agents. Provides control over multiple Oracle Enterprise Communications Broker synchronization processes, including defining applicable Oracle Enterprise Communications Brokers and initiating the synchronization.
- SIP Registrar—Create and manage a SIP registrar object on the Oracle Enterprise Communications Broker to offload Agent of Record registration processes from other network elements.
- LDAP—Define servers and server access rules for using an external LDAP database as a source for user authentication and routing procedures.
- HMR—Create header manipulation rules that change session service messages for interoperability, policy, and other deployment purposes.
- Security—Configure login authentication, certificate records, and TLS profiles. Generate certificate requests and import certificates. Add a public key. Enable audit logging.
- Accounting—Configure connections to RADIUS servers to collect Call Detail Records (CDR) generated by the system.
- SNMP—Specify SNMP community for allowing access to READ functions and trap receivers.
- Web Server—Specify web server functionality, including HTTP and HTTPS operation. Specify the applicable TLS profile and inactivity timeout.

Save and Activate

The Web GUI retains configuration changes until you send them to your device or discard them from the GUI. Configuration dialogs include an "OK" button that sends your changes to the device.

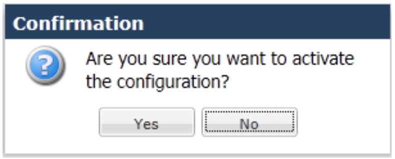
Bear in mind that you must also Save, then Activate your changes before your device actually uses your changes. The Save link, appearing as a disc icon towards the top left corner of each Web GUI page, initiates configuration Save and Activate procedures to your system.

When you click Save, the Web GUI either saves the configuration to your device or prevents you from saving invalid data. The system highlights any fields containing invalid data, allowing you to easily find and correct the mistake.



Saving configuration...

After the save is complete, the Web GUI provides you with a dialog box asking you if you wish to activate this configuration.

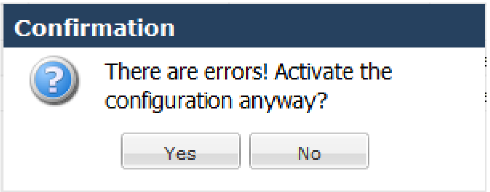


Are you sure you want to activate the configuration?

You are able to perform the save without activation, if desired. This would be common for configuration changes that need to be activated within a preferred window to avoid any service disruption.

The dialog above defaults to “No”, which leaves your changes saved to your system, but not activated. Select No if you want to activate your configuration at a later time. Select Yes to activate. The Web GUI provides a final message box indicating success when it is finished.

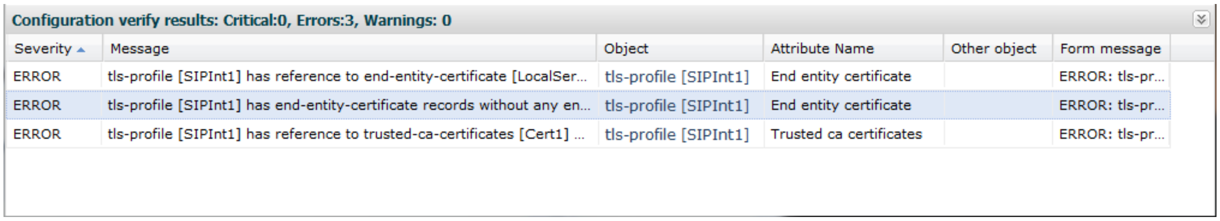
The Web GUI also checks your configuration for errors every time you click the Save button, indicating when it finds them prior to activation. When it discovers configuration errors, the system displays the following dialog.



Confirmation

There are errors! Activate the configuration anyway?

The system displays configuration errors in a list at the bottom of the Web GUI. You can hide and size this error list, an example of which is displayed below. The Web GUI allows you to navigate to the each object in the list by clicking the object in the Object column.



Configuration verify results: Critical:0, Errors:3, Warnings: 0

Severity ▲	Message	Object	Attribute Name	Other object	Form message
ERROR	tls-profile [SIPInt1] has reference to end-entity-certificate [LocalSer...	tls-profile [SIPInt1]	End entity certificate		ERROR: tls-pr...
ERROR	tls-profile [SIPInt1] has end-entity-certificate records without any en...	tls-profile [SIPInt1]	End entity certificate		ERROR: tls-pr...
ERROR	tls-profile [SIPInt1] has reference to trusted-ca-certificates [Cert1] ...	tls-profile [SIPInt1]	Trusted ca certificates		ERROR: tls-pr...

General and System-Config Settings

Use the General icon on the Configuration tab to reach the General and System-Config pages, where you can set the following system-wide parameters.

General

Use **General** to specify the following:

- Network Time Protocol (NTP) servers—Add one or more NTP servers.
- Denial of Service (DoS)—Set the maximum SIP packet and ARP packet rates.
- High Availability (HA)—Enable and disable HA, identify the primary and secondary devices, and specify synchronization.

System-Config

Use **System Config** to specify the following:

- System settings—Set the hostname, location, and default gateway, console timeout, and restart.
- SNMP—Enable and disable SNMP, specify the MIB system, and set SNMP traps and notifications.
- Syslog servers—Add one or more Syslog servers, specify the system log level, and specify the process log level.
- Communications Monitoring Probe—Enable and disable the Communications Monitor, set the group ID, set the TLS profile, enable and disable QoS, and add one or more Monitor collectors.
- Alarm threshold—Set the thresholds for one or more types of alarms.

Configure an NTP Server

You can specify one or more Network Time Protocol (NTP) servers for the Oracle Enterprise Communications Broker (ECB) from the General page.

Note: The ECB media interface does not support management traffic for NTP. When configuring connectivity to these resources, do not configure these resources within a media interface subnet range.

1. Access the System Settings configuration object.

   Configuration > **General** > **General**.
2. On the Modify System settings page, for NTP servers, click **Add**, and enter the address or FQDN for the NTP server that you want to add.

3. (Optional) Add another NTP server to the list.
4. Click **OK** to exit the **Add** dialog.
5. On the Modify Settings page, click **OK**.
6. Save the configuration.

Denial of Service Settings

DoS protection on the Oracle Enterprise Communications Broker employs a means of measuring and limiting traffic based on whether the traffic is SIP signaling or ARP. This categorization aligns with queues that logically separate these traffic types, allowing a simple method of specifying limits. The means by which traffic is defined as trusted or untrusted is, in contrast, a complex set of rule that are not configurable via the GUI.

Configure Denial of Service

Set the SIP and ARP packet rate parameters to configure Denial of Service (DoS) functionality.

1. Access the System Settings configuration object.

   Configuration > **General** > **General**.
2. On the Modify System settings page, expand **Denial of Service Settings**, and do the following:

Attributes	Instructions
Maximum SIP packet rate	Enter the maximum SIP packet rate, in packets per seconds. Range: 20-200,000. Default: 1000.
Maximum ARP packet rate	Enter the maximum ARP packet rate, in packets per seconds. Range: 20-10,000. Default: 10,000.

3. Click **OK**.
4. Save the configuration.

High Availability Settings

High availability is best configured using the ACLI's SETUP wizard. If you use setup, you find the HA fields available from the GUI already configured by SETUP.

Oracle Enterprise Communications Brokers can be deployed in pairs to deliver high availability (HA). Two Oracle Enterprise Communications Brokers operating in this way are called an HA node. Over the HA node, call state is shared, keeping sessions/calls from being dropped in the event of a failure.

Two Oracle Enterprise Communications Brokers work together in an HA node, one in active mode and one in standby mode.

- The active Oracle Enterprise Communications Broker checks itself for internal process and IP connectivity issues. If it detects that it is experiencing certain faults, it hands over its role as the active system to the standby Oracle Enterprise Communications Broker in the node.
- The standby Oracle Enterprise Communications Broker is the backup system, fully synchronized with active Oracle Enterprise Communications Broker’s session status. The standby Oracle Enterprise Communications Broker monitors the status of the active system so that, if needed, it can assume the active role without the active system having to instruct it to do so. If the standby system takes over the active role, it notifies network management using an SNMP trap.

Refer to the *Oracle Enterprise Session Border Controller Configuration Guide* for more detail about High Availability operations, including:

- Synchronization
- Checkpointing

Overview

To produce seamless switchovers from one Oracle Enterprise Communications Broker to the other, the HA node uses shared virtual MAC and virtual IP addresses for the media interfaces in a way that is similar to VRRP (virtual router redundancy protocol). Sharing addresses eliminates the possibility that the MAC and IPv4 address set on one Oracle Enterprise Communications Broker in an HA node will be a single point of failure. The standby Oracle Enterprise Communications Broker sends ARP requests using a utility IPv4 address and its hard-coded MAC addresses to obtain Layer 2 bindings.

When there is a switchover, the standby Oracle Enterprise Communications Broker issues gratuitous ARP messages using the virtual MAC address, establishing that MAC on another physical port within the Ethernet switch. To the upstream router, the MAC and IP are still alive, meaning that existing sessions continue uninterrupted.

Within the HA node, the Oracle Enterprise Communications Brokers advertise their current state and health to one another in checkpointing messages; each system is apprised of the other’s status. Using Oracle’s HA protocol, the Oracle Enterprise Communications Brokers communicate with UDP messages sent out and received on the interfaces carrying "heartbeat" traffic between the active and standby devices.

The standby Oracle Enterprise Communications Broker assumes the active role when:

- It has not received a checkpoint message from the active Oracle Enterprise Communications Broker for a certain period of time.
- It determines that the active Oracle Enterprise Communications Broker’s health score has decreased to an unacceptable level.
- The active Oracle Enterprise Communications Broker relinquishes the active role.

Establishing Active and Standby Roles

Oracle Enterprise Communications Brokers establish active and standby roles in the following ways.

- If a Oracle Enterprise Communications Broker boots up and is alone in the network, it is automatically the active system. If you then pair a second Oracle Enterprise Communications Broker with the first to form an HA node, then the second system to boot up will establish itself as the standby automatically.
- If both Oracle Enterprise Communications Brokers in the HA node boot up at the same time, they negotiate with each other for the active role. If both systems have perfect health, then the Oracle Enterprise Communications Broker with the lowest HA interface IPv4 address will become the active Oracle Enterprise Communications Broker. The Oracle Enterprise Communications Broker with the higher HA interface IPv4 address will become the standby Oracle Enterprise Communications Broker.

If the physical link between the two Oracle Enterprise Communications Brokers fails during boot up or operation, both will attempt to become the active Oracle Enterprise Communications Broker. In this case, processing will not work properly.

Configure High Availability

The Oracle Enterprise Communications Broker (ECB) supports configuring a pair of ECBs for High Availability (HA) operations.

Set the following parameters to configure HA operations.

Note: The ECB automatically populates the Name of primary ECB and Name of secondary ECB fields with the peer names that you entered when you ran the Installation Wizard.

1. Access the System Settings configuration object.

   Configuration > **General** > **General**.

2. On the Modify System settings page, expand **High Availability Settings**, and do the following:

Attributes	Instructions
Enable High Availability	Select to enable HA.

Attributes	Instructions
Name of primary ECB	Enter the name of the primary peer. Default: <primary peer name>.
IP address of primary ECB	Enter the IP address of the primary peer. Default: 169:254.1.1.
Name of secondary ECB	Enter the name of the secondary peer. Default: blank.
IP address of secondary ECB	Enter the IP address of the secondary peer. Default: 169:254.1.2.
Becoming standby time	Enter the time, in milliseconds, to wait for complete synchronization. Range: 5-2147483647. Default: 180,000.
Sync complete time	Enter the timeout, in milliseconds, for subsequent redundancy configuration synchronization requests. Range: 0-4294967295. Default: 1,000.
Sync number transactions	Enter the maximum number of redundancy synchronization transactions to keep. Range: 0-4294967295. Default: 10,000.

3. Click **OK**.
4. Save the configuration.

Forcing an HA Switchover

The Oracle Enterprise Communications Broker allows the user to cause an HA switchover manually. Executing this procedure forces the two Oracle Enterprise Communications Brokers in your HA node to trade roles. The active system becomes standby, and the standby becomes active.

To perform a successful manual switchover, the following conditions must be met:

- The Oracle Enterprise Communications Broker from which you trigger the switchover must be in one of the following states: active, standby, or becoming standby.
- A manual switchover to the active state is only allowed on a Oracle Enterprise Communications Broker in the standby or becoming standby state if it has achieved full media, signaling, and configuration synchronization.
- A manual switchover to the active state is only allowed on a Oracle Enterprise Communications Broker in the standby or becoming standby state if it has a health score above the value you configure for the threshold.

1. Click the **System** tab.
   The Oracle Enterprise Communications Broker displays the system navigation panel to the left of the window displaying the associated controls.
2. Click the **System** tab's **Force HA switchover** link.
   The Oracle Enterprise Communications Broker displays the **Force HA switchover** dialog, which includes a **Switch to standby** button.
3. Click the **Switch to standby** button.
   The Oracle Enterprise Communications Broker executes the HA role change.

Configure System Config

The Oracle Enterprise Communications Broker (ECB) allows you to specify system identification and global settings by way of the parameters that you specify on the System Config page.

Set the following parameters to configure global system identification information.

1. Access the System Config configuration object.

   Configuration > **General** > **System config**.
2. On the Modify System config page, do the following.

Attributes	Instructions
Hostname	Enter the hostname used to identify the ECB by the software. For example, the IP address for Fully Qualified Domain Name.
Description	Enter a textual description of the ECB for informational purposes.
Location	Enter the location of the ECB for informational purposes. For example, you might include the site name and physical address of the ECB.
Default gateway	Set the default gateway for this ECB for egress traffic with no explicit destination. Default: 0.0.0.0. **Note:** Changing this parameter can cause you to lose connectivity with the ECB GUI. Be prepared to access the ECB console, if you lose connectivity. See the *Oracle Communications Session Border Controller ACLI Configuration Guide* for instructions on setting the default gateway using the ACLI.
Restart	Select to cause the system to restart after a service disruption. Default: enabled.
Ssh timeout	Set the length of time, in seconds, that the system waits for the next command before disconnecting. Default: 0. Range: 0-65535.
Console timeout	Set the length of time, in seconds, that the system waits to terminate an ACLI administrative session due to inactivity. Use 0 to disable console session timeout. Default: 0. Range: 0-65535.

3. Save the configuration.

SNMP Configuration

Use SNMP to support monitoring of devices attached to the network for conditions that warrant administrative attention on the Oracle Enterprise Communications Broker (ECB).

Use the MIB settings for informational purposes. The remainder of the parameters enable SNMP and the specific ECB events that you want reported to the SNMP system.

Note that you configure the SNMP community and the trap receiver settings by way of the SNMP icon.

Configure SNMP Settings

Use System Config to enable SNMP on the Oracle Enterprise Communications Broker (ECB) and to set global SNMP settings.

Note that neither the MIB system name nor the MIB system location that you enter in the following procedure correlate to the name and location fields in System Configuration.

1. Access the System Config configuration object.

   Configuration > **General** > **System config**.

2. On the Modify System config page, do the following.

Attributes	Instructions
MIB system contact	Set the contact information displayed in the ECB MIB transactions. You can enter a textual identification of your company's contact person for the ECB and information about how to contact that person.
MIB system name	Set the identification of this ECB to display in MIB transactions. Use the FQDN.
MIB system location	Set the physical location of this ECB to report in MIB transactions.
SNMP enabled	Select to enable SNMP. Note that you must also enable SNMP, and set a snmp-syslog-level. Default: enabled.
Enable snmp auth traps	Select to enable sending an SNMP trap in response to an unsuccessful authentication attempt. Default: disabled.
Enable snmp syslog notify	Select to enable sending SNMP traps when the system generates an alarm. Default: disabled.
Enable snmp monitor traps	• Select to generate traps with unique trap-IDs for each syslog event. • Deselect to generate a single trap-ID for all events, with different values in the description string. Default: disabled.
Enable snmp environment traps	Select to enable environment monitor traps for main board PROM temperature, CPU voltage, power supplies, and fan speeds. Default: disabled.

3. Save the configuration.

Logging (Syslog)

Logging events is a critical part of diagnosing mis-configurations and optimizing operations. Oracle Enterprise Communications Brokers can send both syslog and process log data to appropriate hosts for storage and analysis.

Overview

The Oracle Enterprise Communications Broker generates two types of logs, syslogs and process logs. Syslogs conform to the standard used for logging servers and processes as defined in RFC 3164.

Process logs are Oracle proprietary logs. Process logs are generated on a per-task basis and are used mainly for debugging purposes. Because process logs are more data inclusive than syslogs, their contents encompass syslog log data when they are sent off box. A special application must be run on a remote server to receive process logs. Please contact your Oracle sales representative directly or calling Oracle Customer support for more information about the process log application.

Syslog and process log servers are both identified by an IPv4 address and port pair.

Process Log Messages

Process log messages are sent as UDP packets in the following format:

```
<file-name>:<log-message>
```

In this format, <file-name> indicates the log filename and <log-message> indicates the full text of the log message as it would appear if it were written to the normal log file.

Add a Syslog Server

The Oracle Enterprise Communications Broker (ECB) requires a connection to at least one Syslog Server to process the log events that the system can generate for diagnosing mis-configurations and for optimizing operations. The ECB supports adding up to eight Syslog servers.

1. Access the System Config configuration object.

   Configuration > **General** > **System config**.
2. On the Modify System config page, under Syslog Servers, click **Add**.
3. In the Add Syslog server dialog, do the following:

Attributes	Instructions
Address	Set the IP address or FQDN of the server to which you want to send Syslog messages from the ECB. Default: 0.0.0.0.
Port	Enter the port number on the Syslog server to which the ECB sends log messages. Range: 0-65535. Default: 514.
Facility	Enter the user-defined facility value sent in every syslog message from the ECB to the syslog server. This value must conform to IETF RFC 3164. Range: 0-99999999. Default: 4.

4. Click **OK**.
5. Save the configuration.

Configure Syslog Settings

Set the following parameters to configure system-wide Syslog and Process log functionality. Oracle recommends that you configure Debug and Trace levels temporarily and only when required because both log levels are verbose and can adversely impact system performance.

1. Access the System Config configuration object.

   Configuration > **General** > **System config**.
2. On the Modify System config page, do the following.

Attributes	Instructions
System log level	Select the severity level from the drop-down list that you want to cause the system to send a syslog trap to the Network Management System. Default: Warning.
Process log level	Select the severity level from the drop-down list that you want to cause the system to send a process trap to the Network Management System. Default: Notice.

3. Click **OK**.
4. Save the configuration.

Enterprise Operations Monitor

As a proactive call monitoring solution, the Oracle Enterprise Operations Monitor (EOM) captures and analyzes all required signaling messages and media from the network, providing full correlation and quality metrics in real time. The EOM enables you to drill down into the captured data for troubleshooting and root-cause analysis of any reported problem related to a user, user group, trunk, network device, or Internet Protocol (IP) address. The Enterprise Operations Monitor Mediation Engine (ME) is the application that collects SIP, DNS, ENUM and protocol message traffic received from one or more EOM probes.

You can configure the Oracle Enterprise Communications Broker (ECB) to act as an EOM probe, or as an exporter, that can:

- Establish an authenticated, persistent, reliable TCP connection between itself and one or more Oracle Enterprise Operations Monitor Mediation Engines.
- Send UTC-timestamped, unencrypted copy of a protocol messages to the Oracle Enterprise Operations Monitor Mediation Engine.
- Accompany the copied message with related data to include the port or vlan on which the message was sent and received, the local and remote IP:port information, and the transport layer protocol.

Add a Monitor Collector

You can configure the probes embedded in the Oracle Enterprise Communications Broker (ECB) to establish an IPFIX connection with one or more Oracle Enterprise Operations Monitor Mediation Engines (ME) to collect SIP, DNS, ENUM and protocol message traffic for the Enterprise Operations Monitor (EOM) to analyze. You might want to connect the ECB to multiple MEs, for example, to support monitoring continuity in the event of a service disruption.

- Configure at least one network interface.
- Obtain the IP address and port number of each target Oracle Enterprise Operations Monitor Mediation Engine that you want to connect.

In the following procedure, the Monitor Collector is the ME.

1. Access the System Config configuration object.

   Configuration > **General** > **System config**.
2. On the Modify System settings page, under Monitor Collector, click **Add**, and do the following:

Attributes	Instructions
Address	Set the IP address of the target ME. Default: 0.0.0.0.
Port	Set the port number on which the ME listens. Range: 1025-65535. Default: 4739
Network interface	Select the local network interface from which to export traffic to the ME from the drop-down list. Default: wancom0:0.

3. Click **OK**.
4. (Optional) Repeat steps 2-3 for each additional monitor collector you want to connect to the ECB.
5. Click **OK**.
6. Save the configuration.

Configure Communications Monitoring Probe Settings

Configuring Communications Monitoring Probe settings allows you to make the Oracle Enterprise Communications Broker (ECB) act as a probe, sending network traffic information to an Oracle Communications Session Monitor Mediation Engine.

The Communications Session Monitor is Oracle's Communication Experience Manager. The manager is powered by the Oracle Communications Session Monitor Mediation Engine, a platform that collects SIP, DNS, ENUM, and protocol message traffic received from Oracle Communications Session Monitor Probes. The mediation engine stores the traffic in an internal database, and analyzes aggregated data to provide comprehensive multi-level monitoring, troubleshooting, and interoperability information.

Acting as a Probe, or as an exporter, the ECB can:

- Establish an authenticated, persistent, reliable TCP connection between itself and the Oracle Communications Session Monitor Mediation Engines.
- Send UTC time-stamped, unencrypted copy of a protocol messages to the Mediation Engine.
- Accompany the copied message with related data to include: the port and VLAN on which the message was sent or received, local and remote IP:port information, and the transport layer protocol.

1. Access the System Config configuration object.

   Configuration > **General** > **System config**.
2. Expand **Comm monitor**.

Attributes	Instructions
State	Select to enable the probe.
Sbc grp id	Set the SBC group id parameter to assign an integer value to the ECB in its role as an information exporter. Default: 0.
Monitor collector	Click **Add**, and do the following: 1. Address—Enter the collector IP address to specify the IP address of the target Oracle Communications Session Monitor Mediation Engine. 2. Port—Enter the collector port number of the target Oracle Communications Session Monitor Mediation Engine. Default: 4739. Range: 1025-65535. 3. Network Interface—Select the network interface from which to export traffic to the Oracle Communications Session Monitor Mediation Engine. Most systems use M00:0. 4. Click **OK**. 5. Optional—Repeat to add another monitor collector.

3. Do one of the following:
   - Configure other settings on the Modify System Config page, and click **OK**.
   - Click **Back**.
4. Save the configuration.

Network Interface

The network interface element specifies a logical network interface. The Oracle Enterprise Communications Broker supports only one network interface. You configure a SIP interface and one or more application (SIP) ports over this network interface.

This section explains how to configure a network interface with the GUI. Note that the system initialization procedure creates a network interface. If desired, you can set or change this configuration using the GUI.

Network Interface Configuration

Set the following parameters to configure a network interface.

1. Click the Network icon. The system displays the Modify Network settings dialog.
2. vlan—Enter the identification of a specific virtual interface in a physical interface (e.g., a VLAN tab). If this network interface is not channelized, leave this field blank, and the value will correctly default to **`0`**. The sub-port-id is only required if the operation type is Media. The valid range is:
   - Minimum—0
   - Maximum—4095
3. Network IP address—Enter the IPv4 address of this network interface.
4. Network IP subnet mask—Enter the netmask of this network interface in dotted decimal notation.
5. Network IP gateway address —Enter the gateway that this network interface uses to communicate with the next hop. You can set an additional, secondary gateway via the sec-gateway parameter.
6. DNS server ip address—Enter the IP address of the targeted DNS server.
7. DNS domain—Enter the default domain name.
8. Enable ICMP—See the ensuing section on Enabling ICMP.
9. Enable gateway heartbeat—Within the context of high availability, check this checkbox to allow the network interface to continually confirm that its gateway is reachable.

Enable ICMP

To configure ICMP functionality on a media interface, you define the IPv4 address on your Oracle Enterprise Communications Broker network interface and enable ICMP. Enabling ICMP entries automatically opens the well-known port associated with a service.

Set the following parameters to enable ICMP functionality on a network interface:

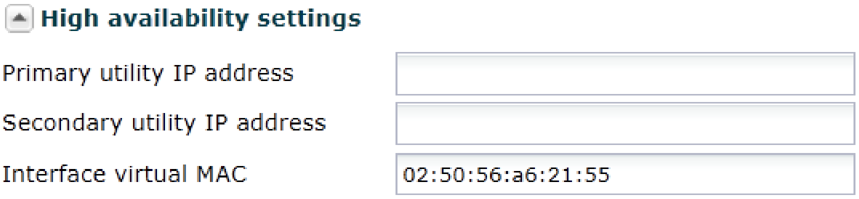
`Enable icmp`—Check the checkbox to enable ICMP on this network interface.

For security and by default, if ICMP is not enabled, the Oracle Enterprise Communications Broker discards ICMP requests or responses for the address. It is recommended that you only enable ICMP temporarily on a network interface.

Network Interface High Availability Configuration

Having configured the first parameters on the Modify Network Settings dialog, the high availability setting fields allow you to manually specify addressing to be used by this interface for high availability operation. It is recommended, however, that you use run setup to configure high availability.

1. Click the arrow next to High Availability settings. The system adds the following fields to the Modify Network Settings dialog.



High availability settings

Primary utility IP address	
Secondary utility IP address	
Interface virtual MAC	02:50:56:a6:21:55

2. Primary utility IP address—Enter the utility IPv4 address for the primary HA peer in an HA architecture. This address can be any unused IPv4 address within the subnet defined for the network interface. For example, given a network interface with the IPv4 address 168.0.4.15/24 (identifying the host associated with the network interface), the possible range of unused IPv4 addresses is 168.0.4.1 to 168.0.4.254. Your network administrator will know which IPv4 addresses are available for use.

3. Secondary utility IP address—Enter the utility IPv4 address for the secondary Oracle Enterprise Communications Broker peer in an HA architecture. Usually, this IPv4 address is the next in the sequence up from the primary utility address. It is also generated from the range of unused IPv4 addresses within the subnet defined for the network interface.

Virtual MAC Addresses

To create an HA node, you create virtual MAC addresses for the media interfaces. You enter these addresses in virtual MAC address parameters for physical interface configurations.

This field is automatically populated with a valid virtual MAC address during run setup. It is recommended that you retain this configuration.

The HA node uses shared virtual MAC (media access control) and virtual IP addresses for the interfaces. When there is a switchover, the standby Oracle Enterprise Communications Broker sends out an ARP message using the virtual MAC address, establishing that MAC on another physical port within the Ethernet switch.

A MAC address is a hardware address that uniquely identifies Oracle Enterprise Communications Broker components. Given that, the virtual MAC address you configure allows the HA node to appear as a single system from the perspective of other network devices. To the upstream router, the MAC and IP are still alive, meaning that existing sessions continue uninterrupted through the standby Oracle Enterprise Communications Broker.

To configure a virtual MAC, enter the virtual MAC address in the **`Interface virtual MAC`** field.

SIP Interface Settings

A SIP Interface is an application layer interface logically residing "over" a network interface. The SIP interface defines the transport addresses (IP address and port) upon which the Oracle Enterprise Communications Broker receives and sends SIP messages. You can define a SIP interface for each network to which the Oracle Enterprise Communications Broker is connected. Note that these networks must be within the Oracle Enterprise Communications Broker's Network Interface subnet. SIP interfaces support UDP, TCP and TLS transport.

In addition to defining a SIP interface's network participation (**Port**), you can also define forking and other functionality (**Interface settings**).

Proxy Registrations

The Oracle Enterprise Communications Broker can proxy registrations when it receives REGISTERs for domains for which it is not a registrar. The user enables this functionality within the **sip-interface**. By default, the Oracle Enterprise Communications Broker rejects the registration.

The Oracle Enterprise Communications Broker's **sip-interface** configuration includes a checkbox titled **Proxy Registrations**, with which the user can enable this function. When checked, the Oracle Enterprise Communications Broker proxies the registration towards the intended registrar. When unchecked, the Oracle Enterprise Communications Broker responds with a **`403: Unauthorized`** message.

Configure a SIP Interface

The SIP interface defines the signalling interface through which the Oracle Enterprise Communications Broker (ECB) receives and sends SIP messages.

Consider any SIP options that you want to add.

In the configuration, you specify how the ECB handles SIP messages and you can add SIP options.

1. Access the SIP Interface configuration object.

   Configuration > **SIP Interface** > **Interface**.
2. On the Modify Interface Settings page, do the following:

Attributes	Instructions
Maximum SIP message length	Set the maximum SIP message length, at which the ECB drops the message. Default: 4096. Range: 0-65535 bytes.
Enable parallel forking	Select the checkbox to cause the system to fork all sessions to all contacts of an Agent of Record.
Enable early media inhibit	Select to extract and store Session Description Protocol (SDP) messages from provisional responses before call setup.
Enable REFER termination	Select to terminate and process SIP REFER messages. Default: Disabled.
Send NOTIFY for REFER provisional responses	Select from the drop-down list which messages to affect .
Fork group timer	Set the timeout value, in seconds, after which the ECB tries the next fork group with the highest priority. Range: 0-32
Default source context	Set the default source context the system uses for a given call when unable to identify source context by way of any other method.
Inbound header manipulation	
Outbound header manipulation	
Enable ToS marking	Select to insert ToS marking for all egress SIP signalling traffic on this interface. Default: Disabled.
ToS value	Enter the RFC 2474 complaint value that you want the ECB to insert in all SIP signalling egress traffic from this interface. Use either a decimal or hexadecimal format.
Stop Recurse	Enter one or more response codes that you want to cause this session agent to stop route recursion. Valid response code values range from 300-599. You can enter individual response codes separated by a comma, such as 301,305 or a range such as 300-380. Default: 401,407.
Proxy registrations	Select to allow the ECB to accept a registration from an unauthorized domain, and proxy the registration to the intended registrar.
SIP options	Click **Add**, enter the option syntax into the dialog, and click either **OK** or **Apply/Add Another**.

3. Click **OK**.
4. Save the configuration.

Restricting Session Initiation

The Oracle Enterprise Communications Broker can restrict the set of end stations that can initiate sessions to those originating via active session agents and previously registered users. By default, the Oracle

Enterprise Communications Broker does not restrict session initiation. The user enables this functionality within the **sip-port**.

The Oracle Enterprise Communications Broker's **sip-port** configuration includes a checkbox titled **Allow session agents and registered end-points** with which the user can restrict session initiation. When checked, the Oracle Enterprise Communications Broker responds to session initiation by endpoints that are not behind an agent or not already registered with a `403: Unauthorized` message.

Configure a SIP Interface Port

A SIP interface port configuration defines the transport address and protocol that the Oracle Enterprise Communications Broker (ECB) uses for sending and receiving messages through a SIP interface. You can apply a TLS profile to the configuration, and you can limit SIP requests from session agents and registered end points. You must configure at least one port per SIP interface. You can optionally configure multiple SIP ports per SIP interface. For example, suppose you configure the ECB to receive calls by way of TCP and to send calls by way UDP, you must configure a SIP port for each protocol.

Configure a TLS profile

In the following procedure, use step 4 to add more SIP interface ports.

1. Access the Ports configuration object. **Configuration tab** > **SIP Interface** > **Port**.
2. On the SIP Ports page, click **Add**, and do the following:

Attributes	**Instructions**
IP address	Enter the IP address of the SIP interface.
IP port	Enter port number for the SIP interface. Default: 5060. Range: 0-65535.
Transport protocol	Select a transport protocol from the drop-down list.
TLS profile	Select a TLS profile from the drop-down list.
Allow session agents and registered endpoints only	Select to allow only session agents and registered endpoints to send a SIP request to the ECB. Default: Disabled.

3. Click **OK**.
   The system displays the SIP Ports page with a list of SIP interface ports you configured.
4. Optional—Click **Add** to add another SIP interface port.
5. Click **Back**.
   The system displays the Configuration tab, where you can do the following:
   - Continue configuring the ECB.
   - Save the configuration.

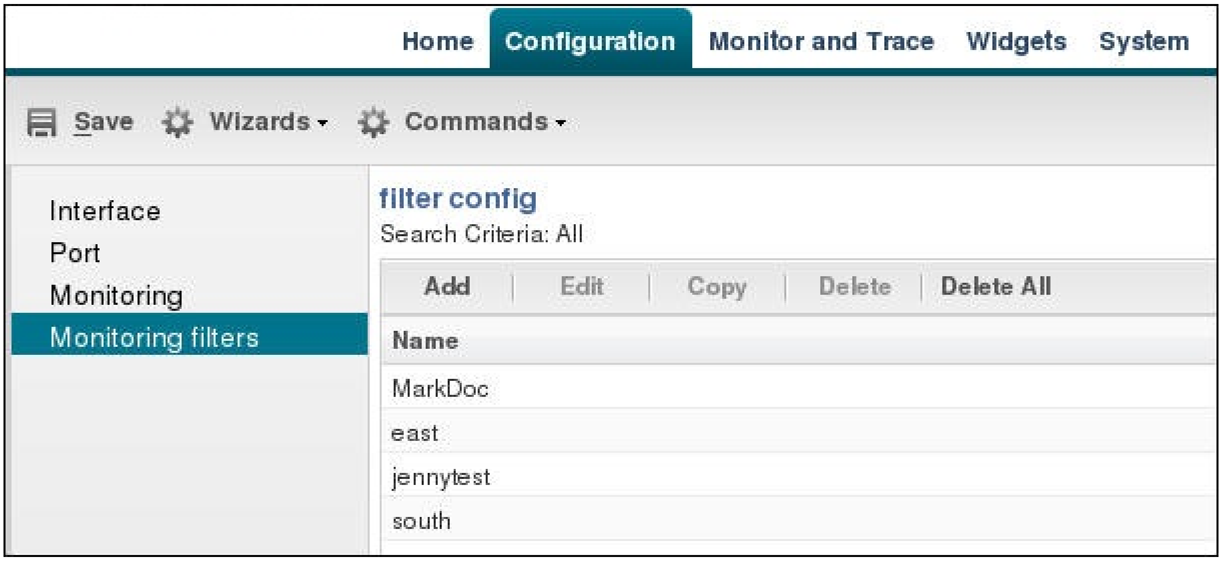
Optional—Configure SIP monitoring.

SIP Monitor and Trace Filter Configuration

The SIP Monitor and Trace function allows you to monitor SIP sessions for notable events and display the results in the Oracle Enterprise Communications Broker (ECB) SIP Notable Events summary. Such information may help you perform troubleshooting. For more targeted monitoring, you can configure filters on particular users and addresses on the ECB, and on a specific agent.

As of PCZ200M4, the ECB includes the following changes:

- The SIP Interface configuration page adds the **Monitoring filters** object to the navigation pane. Use to configure individual filters.

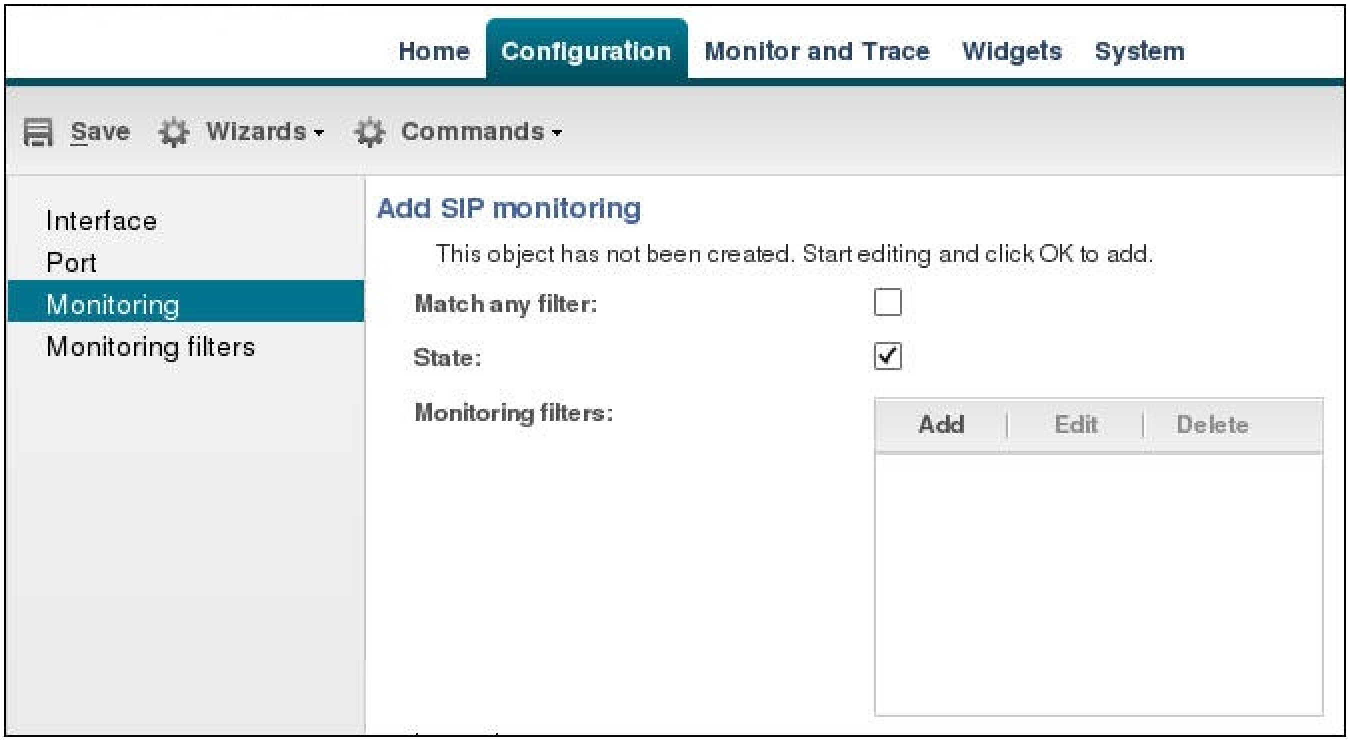


filter config

Search Criteria: All

Name
MarkDoc
east
jennytest
south

- The pre-existing **Monitoring** object on the SIP interface configuration page adds the **Monitoring filters** element to the dialog. Use to apply filters to the ECB.



Add SIP monitoring

This object has not been created. Start editing and click OK to add.

Match any filter: □

State: ☑

Monitoring filters:

- The Add Agents configuration page adds the**Monitoring filters** configuration element to the Advanced section. Use to apply filters to an agent.

- When you upgrade to PCZ200M4, note that the system does not support the former "Enable SIP Monitor and Trace" setting. You must re-configure SNMP event traps through the dialogs described above. See "Caveats" for more information.

Use the following filter configuration process for both new installations and upgrades.

1. Create one or more filters in the Monitoring Filters object. You may use an asterisk character as a filter, if you want to monitor all session data.
2. Add one or more filters to the Monitoring object.
3. (Optional) Add one or more monitoring filters to an agent that you want to monitor.

SIP REFER on the Oracle Enterprise Communications Broker

SIP REFER provides the Oracle Enterprise Communications Broker with the ability to terminate SIP REFER messages and perform attended or unattended call transfers. The user can enable REFER termination at both the agent and SIP interface, with agent configuration taking precedence. In addition the user can configure SIP interface with support for sending NOTIFY messages for provisional responses.

The following sections explain in detail the components of SIP REFER implemented on the Oracle Enterprise Communications Broker.

SIP REFER Method Call Transfer for ECB

The Oracle Enterprise Communications Broker supports a handling mode for the REFER method that automatically converts a received REFER method into an INVITE method. This allows the Oracle Enterprise Communications Broker to transfer a call without having to proxy the REFER back to the other UA.

The Oracle Enterprise Communications Broker has a configuration parameter giving it the ability to provision the handling of REFER methods as call transfers. The parameter is called **Enable REFER termination**. When this feature is enabled, the Oracle Enterprise Communications Broker creates an INVITE message whenever it receives a REFER. The Oracle Enterprise Communications Broker sends this INVITE message to the address in the Refer-To header. Included in the INVITE message is all the unmodified information contained in the REFER message. The previously negotiated SDP is used in the new INVITE message. NOTIFY and BYE messages are sent to the UA upon call transfer completion. The user configures this function at the SIP interface or agent with agent configuration taking precedence.

If a REFER method is received containing no Referred-By header, the Oracle Enterprise Communications Broker adds one, allowing the Oracle Enterprise Communications Broker to support all call agent screen applications.

This SIP REFER method call transfer feature supports the following:

- Both unattended and attended call transfers.
- Both successful and unsuccessful call transfers.
- Early media from the Referred-To party to the transferee.

- REFER method transfer from different sources.
- The REFER event package as defined in RFC 3515. This applies for situations where multiple REFER methods are used within a single dialog.
- Third party initiated REFER method signalling the transfer of a call by associating the REFER method to the dialogue via the REFER TargetDialog.

Unsuccessful Transfer Scenarios

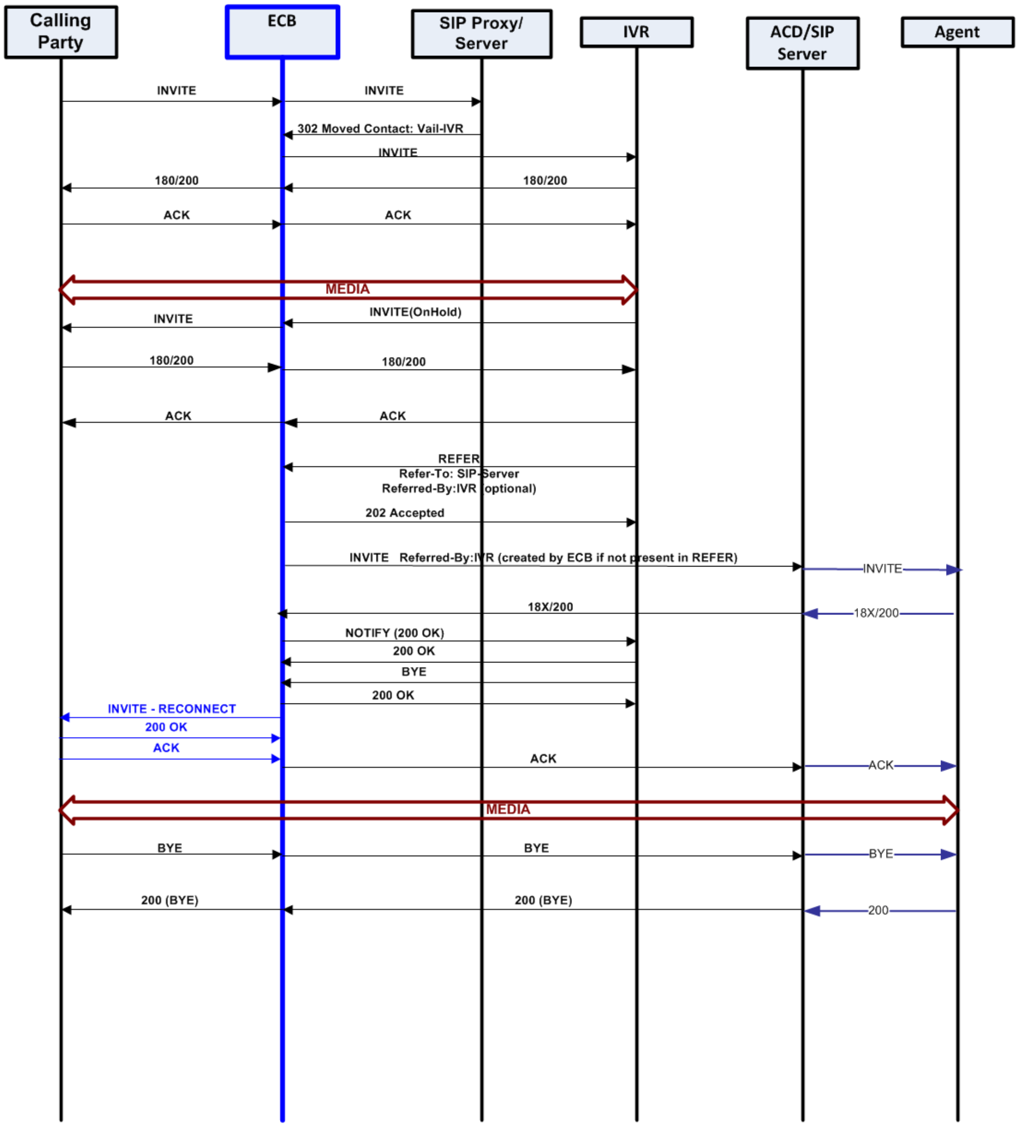
The Oracle Enterprise Communications Broker does not successfully handle the following failed, unusual, and unexpected transfer scenarios:

- The new INVITE to the Referred-To party gets challenged, the Oracle Enterprise Communications Broker does not answer the challenge. It is treated with the 401/407 response just as any other unsuccessful final response.
- The header of the REFER message contains a method other than INVITE or contains URI-parameters or embedded headers not supported by the Oracle Enterprise Communications Broker.
- The Oracle Enterprise Communications Broker shall allow the Referred-To URI that happens to resolve to the same next-hop as the original INVITE went to, to do so.
- The Oracle Enterprise Communications Broker ignores any MIME attachment(s) within a REFER method.
- The Oracle Enterprise Communications Broker recurses (when configured to do so) when the new INVITE sent to the Referred-To party receives a 3xx response.
- The transferee indicated support for 100rel, and the original two parties agreed on using it, yet the Referred-To party does not support it.
- The original parties negotiated SRTP keys.
- The original parties agreed on a codec using a dynamic payload type, and the Referred-To party happens to use a different dynamic payload number for that codec.

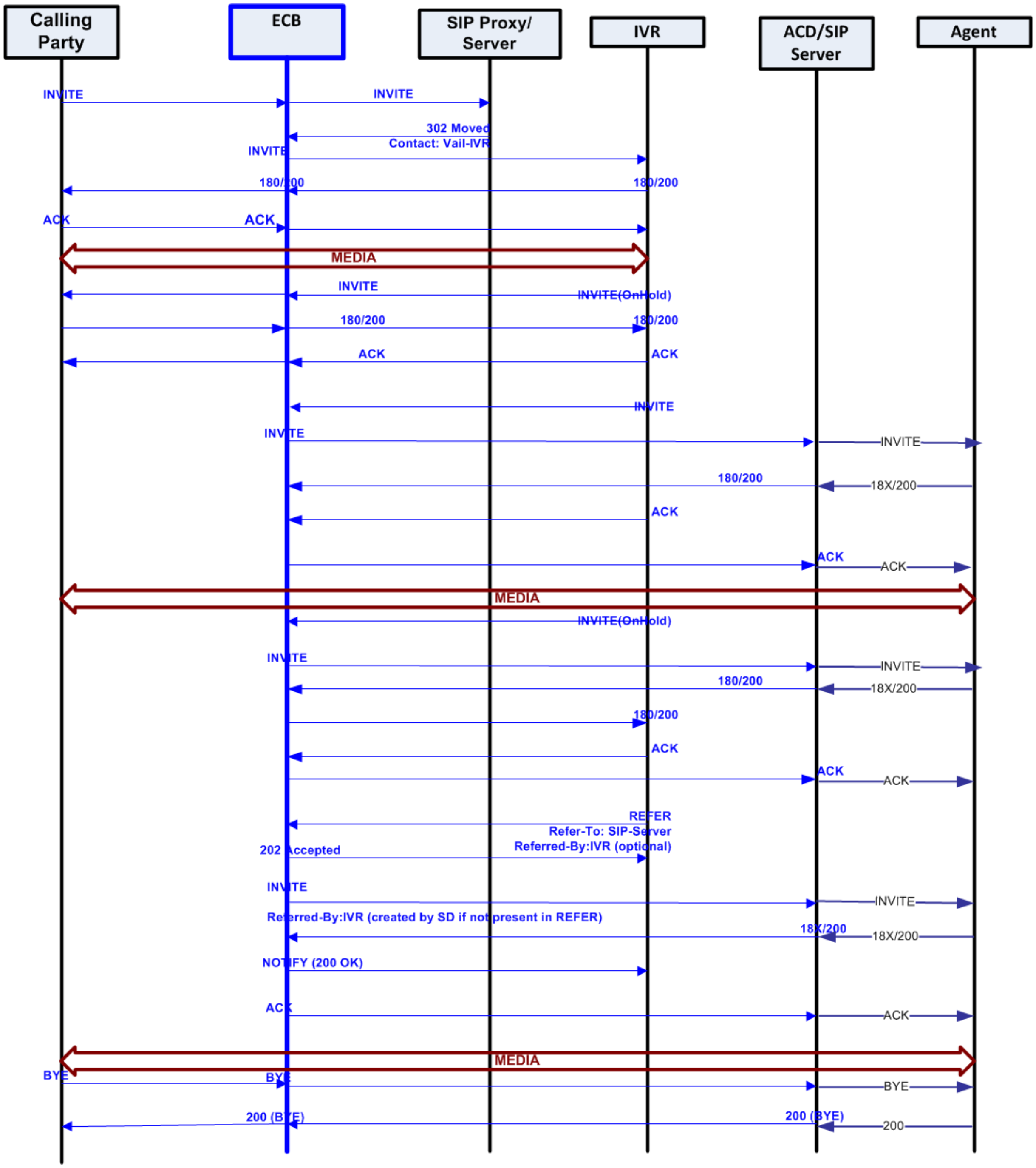
Call Flows

The following is an example call flow for an unattended call transfer:

Initial Configuration



The following is an example call flow of an attended call transfer:



SIP REFER Method Configuration

The Oracle Enterprise Communications Broker allows the user to set REFER termination on a per-agent and/or SIP interface basis. Agent configuration takes precedence over SIP interface configuration.

SIP Interface configuration includes the **Enable REFER termination** checkbox. The user checks the **Enable REFER termination** checkbox to allow this agent to support SIP REFER method call transfers.

Follow the procedure below to enable SIP REFER termination support, setting the SIP interface first, if applicable to your deployment:

1. Navigate to the SIP Interface's SIP REFER configuration fields using the sequence **Configuration** tab > **SIP Interface** icon.
   The Oracle Enterprise Communications Broker displays the **Modify Interface settings** dialog.
2. Locate and check the **Enable REFER termination** checkbox.

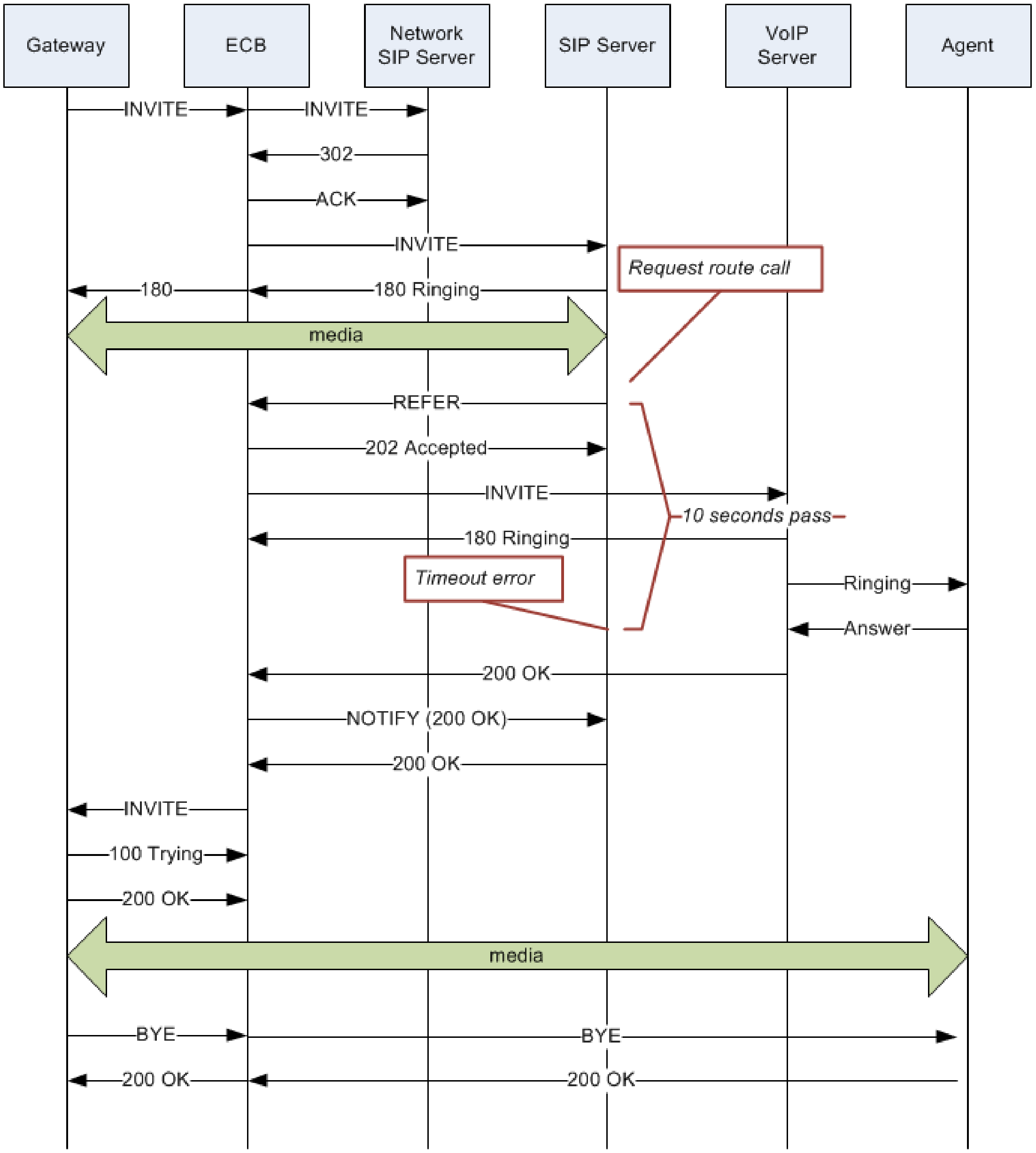
3. Navigate to the SIP REFER configuration fields using the sequence **Configuration** tab > **Agent** icon > **Edit** link.
   The Oracle Enterprise Communications Broker displays the **Modify Agents** dialog.
4. Locate and check the **Enable REFER termination** checkbox.
5. Save and activate your configuration.

180 & 100 NOTIFY in REFER Call Transfers for ECB

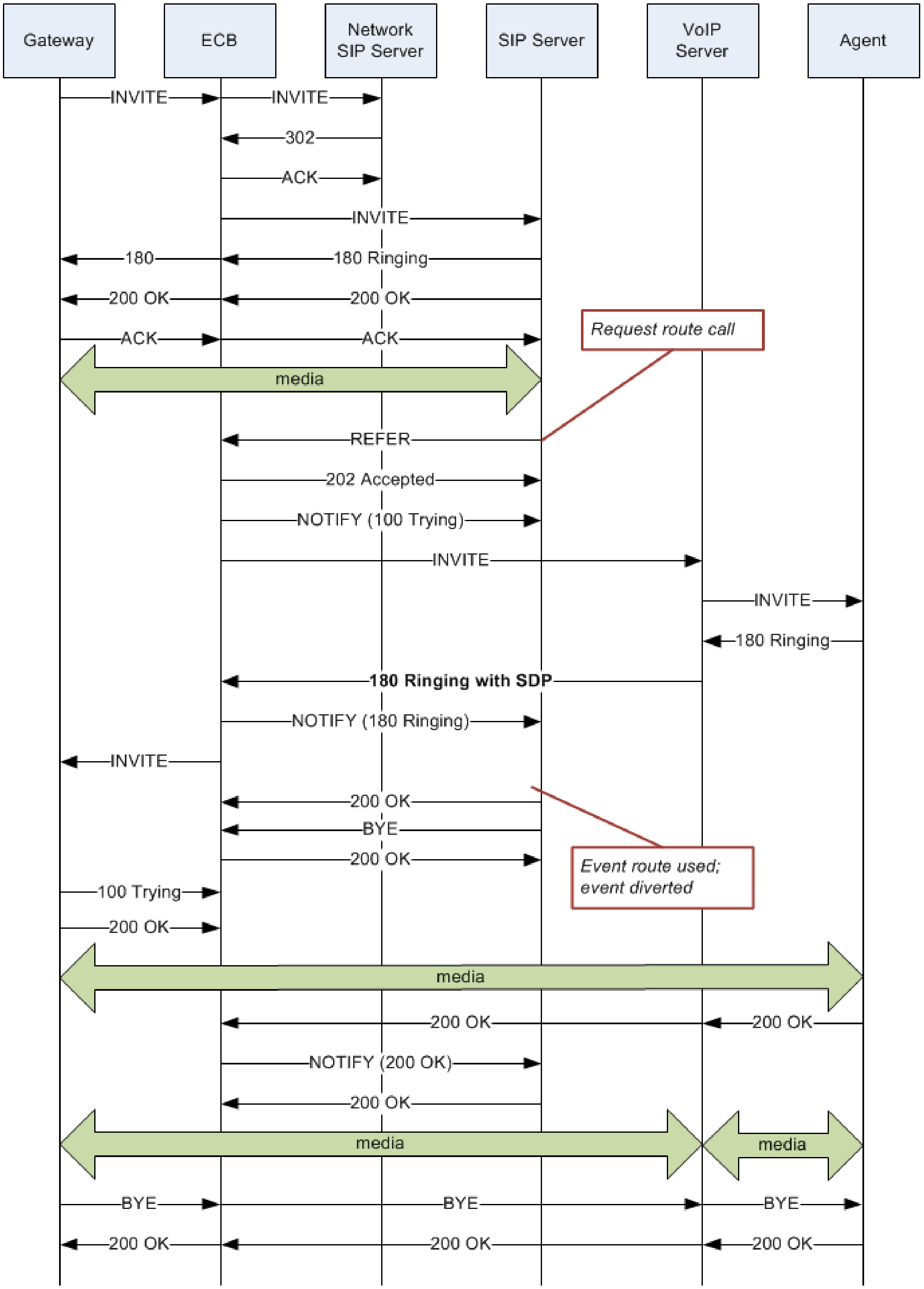
When you configure your Oracle Enterprise Communications Broker to support REFER call transfers, you can enable it to send a NOTIFY message after it has sent either a 202 Accepted or sent a 180 Ringing message. If your network contains elements that comply with RFC 5589, and so expect the NOTIFY message after the 202 Accepted and each provisional 180 Ringing, you want to set the **Send NOTIFY messages for REFER Provisional Responses** to either **`initial`** or **`all`**, according to your needs.

Without this parameter changed from its default (**`none`**), the Oracle Enterprise Communications Broker does not return send the NOTIFY until it receives the 200 OK response from the agent being called. If the time between the REFER and the NOTIFY exceeds time limits, this sequencing can cause the Oracle Enterprise Communications Broker’s NOTIFY to go undetected by devices compliant with RFC 5589. Failures during the routing process can result.

You can see how a sample call flow works without setting the **Send NOTIFY messages for REFER Provisional Responses** parameter.



When you compare the call flow above to the one depicting the scenario when the Oracle Enterprise Communications Broker has the **Send NOTIFY messages for REFER Provisional Responses** changed from its default, you can see that the Oracle Enterprise Communications Broker now response with a NOTIFY in response to the 202 Accepted and it sends another after the 180 Ringing. This causes the event to be diverted successfully.



Sample Messages

In compliance with RFC 5589, the NOTIFY message with 100 Trying as the message body looks like the sample below. Note that the expires value in the subscription state header is populated with a value that equals 2* TIMER C, where the default value of TIMER C is 180000 milliseconds.

```
NOTIFY sips:4889445d8kjtk3@atlanta.example.com;gr=723jd2d SIP/2.0
Via: SIP/2.0/TLS 192.0.2.4;branch=z9hG4bKnas432
Max-Forwards: 70
```

```
To: <sips:transferor@atlanta.example.com>;tag=1928301774
From: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>;tag=a6c85cf
Call-ID: a84b4c76e66710
CSeq: 73 NOTIFY
Contact: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, tdialog
Event: refer
Subscription-State: active;expires=360
Content-Type: message/sipfrag
Content-Length: ...
SIP/2.0 100 Trying
```

Also in compliance with RFC 5589, the NOTIFY message with 180 Ringing as the message body looks like the sample below. Again, the expires value in the subscription state header is populated with a value that equals 2* TIMER C, where the default value of TIMER C is 180000 milliseconds.

```
NOTIFY sips:4889445d8kjtk3@atlanta.example.com;gr=723jd2d SIP/2.0
Via: SIP/2.0/TLS 192.0.2.4;branch=z9hG4bKnas432
Max-Forwards: 70
To: <sips:transferor@atlanta.example.com>;tag=1928301774
From: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>;tag=a6c85cf
Call-ID: a84b4c76e66710
CSeq: 73 NOTIFY
Contact: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, tdialog
Event: refer
Subscription-State: active;expires=360
Content-Type: message/sipfrag
Content-Length: ...
SIP/2.0 180 Ringing
```

Also in compliance with RFC 5589, the NOTIFY message with 200 OK as the message body looks like the sample below.

```
NOTIFY sips:4889445d8kjtk3@atlanta.example.com;gr=723jd2d SIP/2.0
Via: SIP/2.0/TLS 192.0.2.4;branch=z9hG4bKnas432
Max-Forwards: 70
To: <sips:transferor@atlanta.example.com>;tag=1928301774
From: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>;tag=a6c85cf
Call-ID: a84b4c76e66710
CSeq: 74 NOTIFY
Contact: <sips:3ld812adkjw@biloxi.example.com;gr=3413kj2ha>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, tdialog
Event: refer
Subscription-State: terminated;reason=noresource
Content-Type: message/sipfrag
Content-Length: ...
SIP/2.0 200 OK
```

180 and 100 NOTIFY Configuration

You can apply the **Send NOTIFY messages for REFER Provisional Responses** setting to the sip-interface. By default, the Oracle Enterprise Communications Broker only sends the final result NOTIFY message.

To enable 100 and 180 NOTIFY messages in REFER call transfers:

1. Navigate to the SIP Interface's SIP REFER configuration fields using the sequence **Configuration** tab > **SIP Interface** icon.
   The Oracle Enterprise Communications Broker displays the **Modify Interface settings** dialog.
2. Locate the **Send NOTIFY messages for REFER Provisional Responses** and choose from one of the following settings, where the Oracle Enterprise Communications Broker:

- **None**—Disable NOTIFY for REFER provisional responses.
- **initial**—Sends an immediate 100 Trying NOTIFY, and the final result NOTIFY.
- **all**—Sends an immediate 100 Trying NOTIFY, plus a notify for each non-100 provisional messages the Oracle Enterprise Communications Broker receives; and the final result NOTIFY.

3. Save and activate your configuration.

Accounting Settings

The Oracle Enterprise Communications Broker offers support for RADIUS, an accounting, authentication, and authorization (AAA) system. In general, RADIUS servers are responsible for receiving user connection requests, authenticating users, and returning all configuration information necessary for the client to deliver service to the user.

You can configure your Oracle Enterprise Communications Broker to send call accounting information to one or more RADIUS servers. This information can help you to see usage and QoS metrics, monitor traffic, and even troubleshoot your system.

Configuring Accounting

Set the Accounting Configuration parameters in this dialog to indicate where and when you want the system to produce accounting messages.

1. Click the accounting icon. The system displays the Modify Accounting Settings dialog.
2. Enabled—Enable the generation of accounting records by clicking the checkbox or retain the default of disabled.
   - **`enabled | disabled`**
3. Generate Start—Retain the default value ok if you want the CDR Start record to be generated once the system receives an OK message in response to an INVITE. (A CDR Start record informs the accounting server that a SIP session has started.) Other values include:
   - None—Start message should not be generated.
   - Invite—Start message should be generated once the Oracle Enterprise Communications Broker receives a SIP session INVITE.
4. Generate Interim—Retain the default value, Re-invite Response, to cause the Oracle Enterprise Communications Broker to transmit an Interim message. (An Interim message indicates to the accounting server that the SIP session parameters have changed.) Other values include:
   - OK—Start message is generated when the Oracle Enterprise Communications Broker receives an OK message in response to an INVITE.
   - Re-invite—Interim message is generated when the Oracle Enterprise Communications Broker receives a SIP session reINVITE message.
   - Re-invite Cancel—Interim message is generated when the Oracle Enterprise Communications Broker receives a SIP session reINVITE, and the Reinvite is cancelled before the Oracle Enterprise Communications Broker responds to it.
   - Unsuccessful-Attempt—Interim message is generated when a SIP session set-up attempt from a preference-ordered list of next-hop destinations is unsuccessful. The interim message contains: the destination IP address, the disconnect reason, a timestamp for the failure, and the number that was called.
5. Enable file output—Enable the system to generate local files containing accounting records by clicking the checkbox or retain the default of disabled.
   - **`enabled | disabled`**
6. File Path—Specify where, on the system, you want the system to store accounting record files by typing in a valid path.

7. File rotate time—Set how often in minutes you want to rotate the stored files; the Oracle Enterprise Communications Broker overwrites the oldest file first. The minimum rotation time is 2 minutes; the default is 60 minutes. This parameter defaults to 0, and leaving it set to the default means that the Oracle Enterprise Communications Broker does not rotate (or push) the files.
8. Max files—Set the maximum number of files to be stored on the Oracle Enterprise Communications Broker at one time. You can configure the Oracle Enterprise Communications Broker to store as few as one file or as many as 4096. The default is 5.

Configure a RADIUS server to send accounting records (optional).

FTP Push

In addition to local and RADIUS server storage, the Oracle Enterprise Communications Broker provides you with the ability to send accounting files to an FTP server. The information sent to the FTP server is the same as is stored locally.

The FTP push feature is used to copy local CDR files to a remote FTP server on a periodic basis. This feature is configured by defining push receivers which contain standard login and FTP server credentials of the remote machine. At the time interval (file rotate time), the Oracle Enterprise Communications Broker closes the current file and pushes the files that are complete and have not yet been pushed, including the just-closed fille to the FTP server.

Push receiver configurations must include:

- The server’s IP address and port
- Remote path of where to upload the accounting files
- Account login credentials

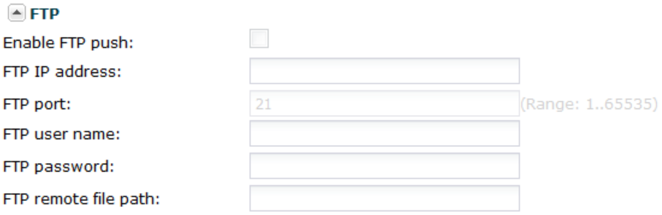
The FTP push configuration creates and pushes accounting files using the following criteria:

- The maximum accounting file size, after which the system creates a new file, is 1000000 bytes.
- The number of files the system creates before it begins to overwrite files (oldest file first) is 5.
- The amount of time between system file push to the FTP server is 60 minutes.

FTP Push Configuration

This configuration assumes a reachable, operating FTP server.

A push receiver configuration includes all the credentials that the Oracle Enterprise Communications Broker needs to log into an FTP server and upload any recent local CDR files. To configure an FTP push server, click the FTP arrow on the Accounting configuration dialog to display the FTP push fields.



FTP

Enable FTP push: □

FTP IP address:

FTP port: 21 (Range: 1..65535)

FTP user name:

FTP password:

FTP remote file path:

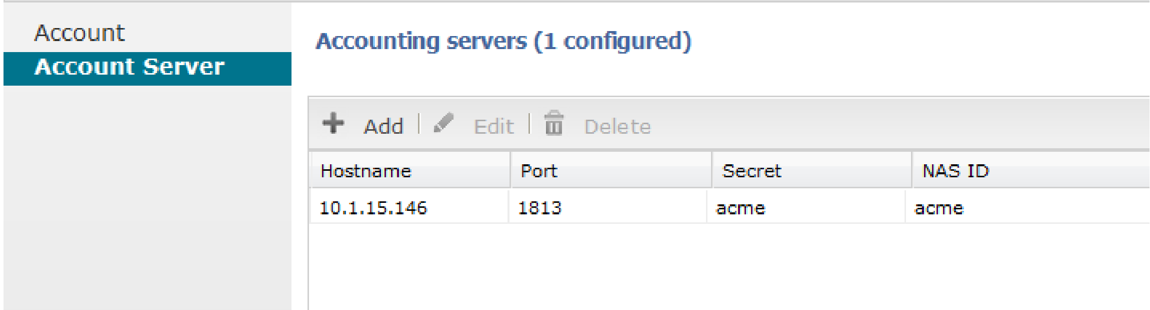
1. Enable FTP push —Check the checkbox to enable FTP push.
2. FTP-address—Set the IP address of this STP server.
3. FTP-port—Set the port of this service:
   - Minimum: 0
   - Maximum: 65535
   - Default: 21
4. FTP-user—Set the username you must use to login to this FTP server.
5. FTP-password—Set the password you must use to login to this FTP server.

6. FTP-remote-path—Set the path on this FTP server on which you want to save your accounting files.

Configuring a RADIUS Account Server

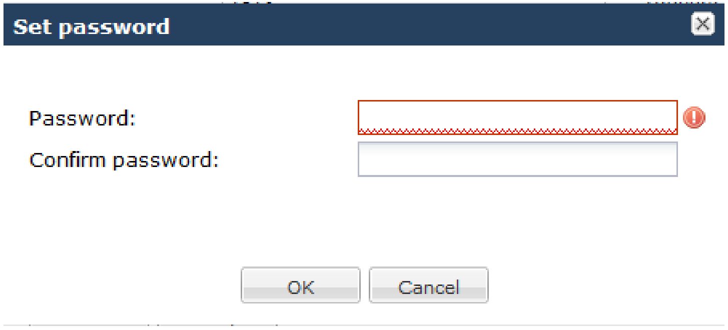
The following procedure is required to provide accounting detail to a RADIUS server.

1. Click the Accounting tab, followed by the Account Server link. The system displays the Add Accounting Server Settings dialog, shown below.



Hostname	Port	Secret	NAS ID
10.1.15.146	1813	acme	acme

2. Click the Add link to add a new server to your list. You can also edit and delete existing servers from links on this dialog.
3. Hostname—Name of the host associated with the account server in hostname format (FQDN) or as an IP address.
4. Port—Retain the default 1813 or enter the number of the UDP port associated with the account server to which RADIUS messages are sent.
   - Minimum: 1025
   - Maximum: 65535
5. Secret—Click the set button. The system displays a password entry and confirm dialog, shown below.



   Enter and then confirm the secret passed from the account server to the client in text format. Transactions between the client and the RADIUS server are authenticated by the shared secret; which is determined by the source IPv4 address of the received packet. You can set or cancel this setting from this dialog using the OK and Cancel buttons respectively.
6. NAS ID—Enter the NAS ID in text format (FQDN allowed). The account server uses this value to identify the Oracle Enterprise Communications Broker for the transmittal of accounting messages. (Optional)
   The remote server to which the account configuration sends messages uses at least one of two potential pieces of information for purposes of identification. The Oracle Enterprise Communications Broker accounting messages always includes in the first of these:
   - Network Access Server (NAS) IP address (the IP address of the Oracle Enterprise Communications Broker's SIP proxy
   - NAS ID (the second piece of information) provided by this value. If you enter a value here, the NAS ID is sent to the remote server.

If you have more than one Oracle Enterprise Communications Broker pointing to the same account server, the NAS ID can be used to identify which Oracle Enterprise Communications Broker generated the record.

Security Settings

Security configuration from the GUI consists of creating the building blocks you can use to establish TLS-secured paths for your signaling traffic. The overall process includes generating certificate requests and certificate import.

The TLS configuration procedures that you can perform from the GUI includes:

- Configure Certificate Records.
- Generate Certificate Request for your CA.
- Import Certificates.
- Upload certificate files.
- Download certificate files.
- Configure TLS Profiles, which utilize your certificate records.
- Apply TLS Profiles to SIP Interfaces, agents and the web-server-config.

The dialogs available from the Security icon allow you to perform all procedures with the exception of applying a TLS profile to a configuration element. You apply TLS profiles to configuration elements using controls within their respective dialogs.

SHA 2 Support

The Oracle Enterprise Communications Broker (ECB) supports Secure Hash Algorithm (SHA) 2 for improved security.

The ECB supports SHA 2 for:

- Generating certificate requests, signing certificates, and verifying certificates.
- Configuring SHA-2 digital certificates on all interfaces through the dashboard, for example, the LDAP, SIP, and web/HTTPS: interfaces.
- Using the 2048 key size as the default for the signing algorithm.
- TLS 1.2 using the SHA-2 algorithm for certificates.

Add a Certificate Record

Use the certificate-record element to add certificate records to the Oracle Enterprise Communications Broker (ECB).

- Confirm that the system displays the Expert mode.

A certificate record represents either the end-entity or the Certificate Authority (CA) certificate on the ECB. When you configure a certificate for the ECB, the name that you enter must be the same as the name that you use to generate a certificate request. If configuring for an end stations CA certificate for mutual authentication, the certificate name must be the same name used during the import procedure.

- If this certificate record is used to present an end-entity certificate, associate a private key with this certificate record by using a certificate request.
- If this certificate record is created to hold a CA certificate or certificate in pkcs12 format, a private key is not required.

1. From the Web GUI, click **Configuration** > **Security** > **Certificate record**.
2. On the Certificate record page, click**Add**.
3. On the Add certificate record page, click **Show advanced**, and do the following:

Attributes	Instructions
Name	Enter the name of the certificate record.
Country	Enter a two character country name abbreviation. For example, US for the United States.
State	Enter a two character state or province name abbreviation. For example, NE for Nebraska.
Locality	Enter the name of the locality in the state or province. For example, a city, a township, or a parish. Range: 1-128 characters.
Organization	Enter the name of the organization holding the certificate. For example, a company name. Range: 1-64 characters.
Unit	Name of the unit within the organization holding the certificate. For example, a business unit or a department. Range: 1-64 characters.
Common name	Common name for the certificate record. For example, your name. Range: 1-64 characters.
Key size	Size of the key for the certificate. Supported values: 512 | 1024 | 2048. Default: 2048 .
Alternate name	Alternate name of the certificate holder.
Trusted	Select to trust this certificate record.
Key usage list	Click **Add** and select a key that you want to use with this certificate record from the drop-down list, and do one of the following: • Click **OK**. • Click **Apply/Add Another**, add another key , and click **OK**. Repeat as needed. This parameter defaults to the combination of digitalSignature and keyEncipherment. For a list of other valid values and their descriptions, see the section “Key Usage Control” in the *ACLI Configuration Guide*.
Extended key usage list	Click **Add**, select an extended key that you want to use with this certificate record from the drop-down list, and do one of the following: • Click **OK**. • Click **Apply/Add Another**, add another extended key, and click **OK**. Repeat as needed. This parameter defaults to serverAuth. For a list of other valid values and their descriptions, see the section “Key Usage Control” in the *ACLI Configuration Guide*.
Options	

4. Click **OK**.
5. Save the configuration.

- Create TLS profiles, using the certificate records to further define the encryption behavior and to provide an entity that you can apply to a SIP interface.

TLS Profile Configuration

Certificate records must exist prior to this configuration.

Configure a TLS profile to further define the encryption behavior you want between these systems and to establish an entity that you can apply to SIP Interfaces. Steps required follow.

1. Click the TLS Profile link. The system displays the TLS profile list.
2. Click the Add link. The system displays the dialog below, which is truncated for the purpose of presentation here.
3. Name—Enter the name of the TLS profile. This parameter is required.
4. end-entity-certificate—Enter the name of the Certificate Record for the applicable entity.
5. trusted-ca-certificates—Enter the names of the trusted CA certificate records.
6. cipher-list—The following cipher-lists are supported for the GUI only:
    - AES256-SHA (TLS_RSA_WITH_AES_256_CBC_SHA) - Firefox (version 12) and Chrome (version 19.0.1084.46m)
    - AES128-SHA (TLS_RSA_WITH_AES_128_CBC_SHA) - Firefox (version 12) and Chrome (version 19.0.1084.46m)
    - DES-CBC-SHA (SSL_RSA_WITH_DES_CBC_SHA or TLS_RSA_WITH_DES_CBC_SHA) - Internet Explorer (Version 9)
7. verify-depth—Specify the maximum depth of the certificate chain that will be verified. The default value is 10. The valid range is:
    - Minimum-0
    - Maximum-10
8. mutual-authenticate—Define whether or not you want the Oracle Enterprise Communications Broker to mutually authenticate the client. The default value is disabled. The valid values are:
    - enabled-disabled (default)
9. tls-version—Enter the TLS version you want to use with this TLS profile. Default is compatibility. Valid values are:
    - TLSv1
    - SSLv3
    - compatibility (default)
10. cert-status-check—Enables OCSP in conjunction with an existing TLS profile.
11. cert-status-profile-list—Assigns one or more cert-status-profiles to the current TLS profile. Each assigned cert-status-profile provides the information needed to access a single OCSP responder.
12. ignore-dead-responder—Enables your device to establish a client connection even if the OCSP responder is unavailable, assuming the associated certificate was signed by a trusted certificate authority.
    - enabled-disabled (default)
13. allow-self-signed-cert—Enables your device to establish client connections to clients that present self-signed certificates.
    - enabled-disabled (default)

Apply your TLS profile to a SIP Interface by selecting if from the SIP Interface's TLS Profile drop-down.

Generate a Certificate Request from the GUI

Use the certificate-record element to select a certificate record and generate a certificate request.

- Confirm that the certificate record exists.

To get a certificate authorized by a Certificate Authority (CA), you must generate a certificate request from the certificate record on the device and send it to the CA.

1. From the Web GUI, click **Configuration** > **security** > **certificate-record**.
   The system displays a list of certificate records.
2. Select the certificate record for the device.
3. Click **Generate**.
   The system creates the request and displays it in a dialog.
4. Copy the information from the dialog and send it to your CA as a text file.

- When the CA replies with the certificate, import the certificate to the device with the corresponding certificate record.

Import a Certificate

Use the certificate-record element to import a certificate into the Oracle Enterprise Communications Broker (ECB).

Use this procedure to import either a device certificate or an end-station CA certificate for a mutual authentication deployment. You must import the certificate to the corresponding certificate record for the ECB. End-station CA certificates may or may not need to be imported against a pre-configured certificate record.

1. From the Web GUI, click **Configuration** > **security** > **certificate record** .
2. Select the certificate record for the device.
3. Click **Import**.
   The system displays a dialog from which you can import the certificate.
4. Select one of the following format types from the **Format** drop down list:
   - pkcs7
   - x509
   - Try-all. The system tries all possible formats until it can import the certificate.
5. Browse to the certificate file, and select the certificate to import.
6. Click **Import**.
   The system imports the certificate.

- Apply the corresponding certificate record to the intended SIP interface.

RADIUS Authentication

The User Authentication and Access control feature supports authentication using one or more RADIUS servers. In addition, you can set two levels of privilege, one for all privileges and more limited set that is read-only.

User authentication configuration also allows you to use local authentication, localizing security to the Oracle Enterprise Communications Broker (ECB) log-in modes. These modes are User and Superuser, each requiring a separate password.

The components involved in the RADIUS-based user authentication architecture are the ECB and your RADIUS servers. In these roles:

- The ECB restricts access and requires authentication through the RADIUS server. The ECB communicates with the RADIUS server using either port 1812 or 1645, but does not know whether or not the RADIUS server listens on these ports

- Your RADIUS server provides an alternative method for defining ECB users and authenticating them through RADIUS. The RADIUS server supports the VSA called ACME_USER_CLASS, which specifies what kind of user is requesting authentication and what privileges to grant.

  The ECB also supports the use of the Cisco Systems Inc.™ Cisco-AVPair vendor specific attribute (VSA). This attribute allows for successful administrator login to servers that do not support the Oracle authorization VSA. While using RADIUS-based authentication, the ECB authorizes you to enter Superuser mode locally even when your RADIUS server does not return the ACME_USER_CLASS VSA or the Cisco-AVPair VSA. For this VSA, the Vendor-ID is 1 and the Vendor-Type is 9. The following below shows the values this attribute can return, and the result of each:

  - shell:priv-lvl=15—User automatically logged in as an administrator
  - shell:priv-lvl=1—User logged in at the user level, and not allowed to become an administrator
  - Any other value—User rejected

When RADIUS user authentication is enabled, the ECB communicates with one or more configured RADIUS servers that validates the user and specifies privileges. On the ECB, you configure:

- What type of authentication you want to use on the ECB
- If you are using RADIUS authentication, you set the port from which you want the ECB to send messages
- If you are using RADIUS authentication, you also set the protocol type you want the ECB and RADIUS server to use for secure communication

Although most common deployments use two RADIUS servers to support this feature, you may configure up to six. Among other settings for the server, there is a class parameter that specifies whether the ECB should consider a specific server as primary or secondary. As implied by these designations, the primary servers are used first for authentication, and the secondary servers are used as backups. If you configure more than one primary and one secondary server, the ECB chooses servers to which it sends traffic in a round-robin strategy. For example, if you specify three servers are primary, the ECB will round-robin to select a server until it finds an appropriate one. The system does the same for secondary servers.

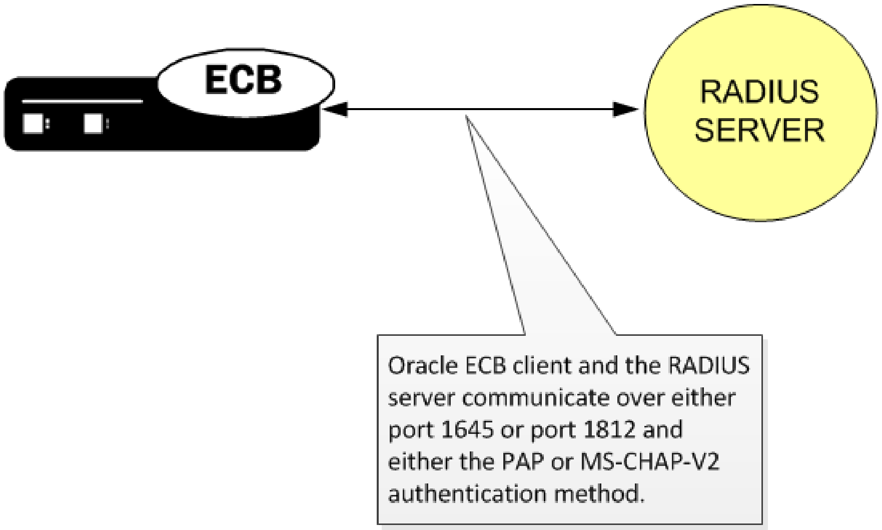
The VSA attribute assists with enforcement of access levels by containing one of the following classes:

- None—All access denied
- User—Monitoring privileges are granted; your user prompt will resemble ORACLE>
- Admin—All privileges are granted (monitoring, configuration, etc.); your user prompt will resemble ORACLE#

After the system selects a RADIUS server, the ECB initiates communication and proceeds with the authentication process. The authentication process between the ECB and the RADIUS server takes place uses one the following methods, all of which are defined by RFCs:

Protocol	RFC
PAP (Password Authentication Protocol)	B. Lloyd and W. Simpson, PPP Authentication Protocols, RFC 1334, October 1992
CHAP (Challenge Handshake Authentication Protocol)	B. Lloyd and W. Simpson, PPP Authentication Protocols, RFC 1334, October 1992 W. Simpson, PPP Challenge Handshake Authentication Protocol (CHAP), RFC 1994, August 1996
MS-CHAP-V2	G. Zorn, Microsoft PPP CHAP Extensions, Version 2, RFC 2759, January 2000

Note: MS-CHAP-V2 support includes authentication only; password exchange is not supported or allowed on the ECB.



Oracle ECB client and the RADIUS server communicate over either port 1645 or port 1812 and either the PAP or MS-CHAP-V2 authentication method.

Management Protocol Behavior

When you use local authentication, management protocols behave the same way that they do when you are not using RADIUS servers. When you use RADIUS servers for authentication, management protocols behave as follows:

- SSH in pass-through mode—The User and Admin accounts are authenticated locally, not through the RADIUS server. For all other accounts, the configured RADIUS servers are used for authentication. When authentication is successful, the user is granted privileges depending on the ACME_USER_CLASS VSA attribute.
- SSH in non-pass-through mode—When you create an SSH account on the Oracle Enterprise Communications Broker (ECB), you are asked to supply a user name and password. When local authentication succeeds, you are prompted for the ACLI user name and password. If your user ACLI name is user, then you are authenticated locally. Otherwise, you are authenticated using the RADIUS server. If RADIUS authentication is successful, the privileges you are granted depend on the ACME_USER_CLASS VSA attribute.
- SFTP in pass-through mode—When you do not configure an SSH account on the Oracle Enterprise Communications Broker, the RADIUS server is contacted for authentication for any user that does not have the user name user. The Oracle Enterprise Communications Broker uses local authentication if the user name is user.
- SFTP in non-pass-through mode—The User and Admin accounts are authenticated locally, not through the RADIUS server. For all other accounts, the configured RADIUS servers are used for authentication.

RADIUS Authentication Configuration

To enable RADIUS authentication and user access on your Oracle Enterprise Communications Broker, you need to configure global parameters for the feature and then configure the RADIUS servers that you want to use.

Global Authentication Settings

To configure the global authentication settings:

1. Click the **Configuration** tab.
   The Oracle Enterprise Communications Broker displays the configuration panel.
2. Click the **Security** configuration icon.
   The Oracle Enterprise Communications Broker displays the security configuration panel.
3. Click the **Login authentication** link from the navigation panel on the left-hand side of the security configuration panel.
   The Oracle Enterprise Communications Broker displays the **Modify Authentication** dialog.

4. Set the number of the port you want to use from message sent from the Oracle Enterprise Communications Broker to the RADIUS server in the **Source port** field. The default value is 1812. The valid values are:
   - 1645 | 1812
5. Set the type of user authentication you want to use on this Oracle Enterprise Communications Broker using the **Type** drop-down list. The default value is **`local`**. The valid values are:
   - local | radius
6. If you are using RADIUS user authentication, set the protocol to use with your RADIUS server(s) from the **Protocol** drop-down list. The default is **`pap`**. The valid values are:
   - pap | chap | mschapv2
7. Set the **allow-local-authorization parameter** to **`enabled`** if you want the Oracle Enterprise Communications Broker to authorize users to enter Superuser (administrative) mode locally even when your RADIUS server does not return the ACME_USER_CLASS VSA or the Cisco-AVPair VSA. The default for this parameter is **`disabled`**.
8. Check the **Login as admin** checkbox if you want users to be logged automatically in Superuser (administrative) mode. The default for this parameter is disabled.

RADIUS Server Settings

The parameters you set for individual RADIUS servers identify the RADIUS server, establish a password common to the Oracle Enterprise Communications Broker and the server, and establish trying times.

Setting the class and the authentication methods for the RADIUS servers can determine how and when they are used in the authentication process.

To configure a RADIUS server to use for authentication:

1. Navigate to the Radius servers list box directly below the main authentication configuration controls. The list box displays all previously configured Radius servers, if any. You can Add, Edit, Copy and Delete existing servers using the control across the top of this list box.
2. Click the Add link.
   The Oracle Enterprise Communications Broker displays the Add Radius server dialog.
3. Set the remote IP address for the RADIUS server in the **Add** field. There is no default value, and you are required to configure this address.
4. Set the port at the remote IP address for the RADIUS server in the **Port** field. The default port is set to **`1812`**. The valid values are:
   - 1645 | 1812
5. Set the state of the RADIUS server in the **State** field. Enable this parameter to use this RADIUS server to authenticate users. The default value is **`enabled`**. The valid values are:
   - enabled | disabled
6. Set the password that the RADIUS server and the Oracle Enterprise Communications Broker share in the **secret** dialog, available when you click the **set** button. This dialog requires you to enter the secret twice and click **OK**. This password is transmitted between the two when the request for authentication is initiated; this ensures that the RADIUS server is communicating with the correct client.
7. Set the NAS ID for the RADIUS server in the **Nas id** field. There is no default for this parameter.
8. Set the number of times that you want the Oracle Enterprise Communications Broker to retry for authentication information from this RADIUS server in the **retry-limit** field. The default value is **`3`**. The valid range is:
   - Minimum—1
   - Maximum—5

   If the RADIUS server does not respond within this number of tries, the Oracle Enterprise Communications Broker marks is as dead.

9. Set the amount of time (in seconds) that you want theOracle Enterprise Communications Broker to wait before retrying for authentication from this RADIUS server in the **retry-time** field. The default value is `5`. The valid range is:
   - Minimum—5
   - Maximum—10
10. Set the amount of time in seconds before the Oracle Enterprise Communications Broker retries a RADIUS server that it has designated as dead because that server did not respond within the maximum number of retries in the **dead-time** field. The default is `10`. The valid range is:
    - Minimum—10
    - Maximum—10000
11. Set the maximum number of outstanding sessions for this RADIUS server. The default value is `255` in the **maximum-sessions** field. The valid range is:
    - Minimum—1
    - Maximum—255
12. Set the class of this RADIUS server as either primary or secondary in the **class** field. A connection to the primary server is tried before a connection to the secondary server is tried. The default value is `primary`. Valid values are:
    - primary | secondary

      The Oracle Enterprise Communications Broker tries to initiate contact with primary RADIUS servers first, and then tries the secondary servers if it cannot reach any of the primary ones.

      If you configure more than one RADIUS server as primary, the Oracle Enterprise Communications Broker chooses the one with which it communicates using a round-robin strategy. The same strategy applies to the selection of secondary servers if there is more than one.
13. Set the authentication method you want the Oracle Enterprise Communications Broker to use with this RADIUS server from the in the **authentication-method** drop-down. The default value is `pap`. Valid values are:
    - all | pap | chap | mschapv2

      This parameter has a specific relationship to the global protocol parameter for the authentication configuration, and you should exercise care when setting it. If the authentication method that you set for the RADIUS server does not match the global authentication protocol, then the RADIUS server is not used. The Oracle Enterprise Communications Broker simply overlooks it and does not send authentication requests to it. You can enable use of the server by changing the global authentication protocol so that it matches.
14. Save your work and activate your configuration.

TACACS+ Overview

Like DIAMETER and RADIUS, TACACS+ uses a client/server model in which a Network Access Server (NAS) acts in the client role and a TACACS+ equipped device (a daemon in TACACS+ nomenclature) assumes the server role. For purposes of the current implementation, the Oracle Enterprise Communications Broker functions as the TACACS+ client. Unlike RADIUS, which combines authentication and authorization, TACACS+ provides three distinct applications to provide finer grade access control.

Authentication is the process that confirms a user’s purported identity. Authentication is most often based on a simple username/password association, but other, and more secure methods, are becoming more common. The following authentication methods are support by the current implementation: simple password, PAP (Protocol Authentication Protocol), and CHAP (Challenge Handshake Authentication Protocol).

Authorization is the process that confirms user privileges. TACACS+ can provide extremely precise control over access to system resources. In the current implementation, TACACS+ controls access to system administrative functions.

TACACS+ provides secure communication between the client and daemon by encrypting all packets. Encryption is based on a shared-secret, a string value known only to the client and daemon. Packets are encrypted in their entirety, save for a common TACACS+ header.

The cleartext header contains, among other fields, a version number, a sequence number. and a session ID. Using a methodology described in Section 5 of the TACACS+ draft RFC, the sender encrypts outbound cleartext messages by repetitively running the MD5 hash algorithm over the concatenation of the session ID, shared-secret, version number, and sequence number values, eventually deriving a virtual one-time-pad of the same length as the message body. The sender encrypts the cleartext message with an XOR (Exclusive OR) operation, using the cleartext message and virtual one-time-pad as inputs.

The message recipient, who possesses the shared-secret, can readily obtain the version number, sequence number, session ID, and message length from the cleartext header. Consequently, the recipient employs the same methodology to derive a virtual one-time-pad identical to that derived by the sender. The recipient decrypts the encrypted message with an XOR operation, using the encrypted message and virtual one-time-pad as inputs.

Details on the TACACS+ functions and configuration can be found in the Oracle Communications Session Border Controller ACLI Configuration Guide.

The TACACS+ implementation is based upon the following internet draft.

draft-grant-tacacs-02.txt, *The TACACS+ Protocol Version 1.78*

Other relevant documents include

RFC 1321, *The MD-5 Message Digest Algorithm*

RFC 1334, *PPP Authentication Protocols* .

RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*

TACACS+ Authentication

The Oracle Enterprise Communications Broker uses TACACS+ authentication services solely for the authentication of user accounts. Administrative users must be authenticated locally by the Oracle Enterprise Communications Broker.

The current TACACS+ implementation supports three types of user authentication: simple password (referred to as ascii by TACACS+), PAP, and CHAP.

ascii Login

ascii login is analogous to logging into a standard PC. The initiating peer is prompted for a username, and, after responding, is then prompted for a password.

PAP Login

PAP is defined in RFC 1334, *PPP Authentication Protocols*. This protocol offers minimal security in that passwords are transmitted as unprotected cleartext. PAP login differs from ascii login in that the username and password are transmitted to the authenticating peer in a single authentication packet, as opposed to the two-step prompting process used in ascii login.

CHAP Login

CHAP is defined in RFC 1994, *PPP Challenge Handshake Authentication Protocol.* CHAP is a more secure than PAP in that it is based on a shared-secret (known only to the communicating peers), and therefore avoids the transmission of cleartext authentication credentials. CHAP operations can be summarized as follows.

After a login attempt, the initiator is tested by the authenticator who responds with a packet containing a challenge value — an octet stream with a recommended length of 16 octets or more. Receiving the challenge, the initiator concatenates an 8-bit identifier (carried within the challenge packet header), the shared-secret, and the challenge value, and uses the shared-secret to compute an MD-5 hash over the concatenated string. The initiator returns the hash value to the authenticator, who performs the same hash calculation, and compares results. If the hash values match, authentication succeeds; if hash values differ, authentication fails.

Authentication Message Exchange

All TACACS+ authentication packets consist of a common header and a message body. Authentication packets are of three types: START, CONTINUE, and REPLY.

START and CONTINUE packets are always sent by the Oracle Enterprise Communications Broker, the TACACS+ client. START packets initiate an authentication session, while CONTINUE packets provide authentication data requested by the TACACS+ daemon. In response to every client-originated START or CONTINUE, the daemon must respond with a REPLY packet. The REPLY packet contains either a decision (pass or fail), which terminates the authentication session, or a request for additional information needed by the authenticator.

TACACS+ Header

The TACACS+ header format is as follows.

```
+----+----+--------+--------+--------+
|maj |min | type   | seq_no | flags  |
|ver |ver |        |        |        |
+----+----+--------+--------+--------+
| session_id                         |
+------------------------------------+
| length                             |
+------------------------------------+
```

maj ver

This 4-bit field identifies the TACACS+ major protocol version, and must contain a value of 0xC .

min ver

This 4-bit field identifies the TACACS+ minor protocol version, and must contain either a value of 0x0 (identifying TACACS+ minor version 0) or a value of 0x1 . (identifying TACACS+ minor version 1). Minor versions 0 and 1 differ only in the processing of PAP and CHAP logins.

type

This 8-bit field identifies the TACACS+ AAA service as follows:

0x1 — TACACS+ Authentication

0x2 — TACACS+ Authorization

0x3 — TACACS+ Accounting

sequence-no

This 8-bit field contains the packet sequence for the current session.

The first packet of a TACACS+ session must contain the value 1; each following packet increments the sequence count by 1. As TACACS+ sessions are always initiated by the client, all client-originated packets carry an odd sequence number, and all daemon-originated packets carry an even sequence number. TACACS+ protocol strictures do not allow the sequence_no field to wrap. If the sequence count reaches 255, the session must be stopped and restarted with a new sequence number of 1.

flags

This 8-bit field contains flags as described in Section 3 of the draft RFC; flags are not under user control.

session_id

This 32-bit field contains a random number that identifies the current TACACS+ session — it is used by clients and daemons to correlate TACACS+ requests and responses.

length

This 32-bit field contains the total length of the TACACS+ message, excluding the 12-octet header — in other words, the length of the message body.

Authentication START Packet

The Oracle Enterprise Communications Broker, acting as a TACACS+ client, sends an authentication START packet to the TACACS+ daemon to initiate an authentication session. The daemon must respond with a REPLY packet.

The authentication START packet format is as follows.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x1         |
+--------+--------+--------+--------+
|action  |priv_lvl|authen_ |service |
|        |        |type    |        |
|--------+--------+--------+--------+
|user_len|port_len|rem_addr|data_len|
|        |        |_len    |        |
+--------+--------+--------+--------+
|              user ...             |
+-----------------------------------+
|              port ...             |
+-----------------------------------+
|            rem-addr ...           |
+-----------------------------------+
|              data ...             |
+-----------------------------------+
```

action

This 8-bit field contains an enumerated value that identifies the requested authentication action. For the current TACACS+ implementation, this field always contains a value of 0x01 , indicating user login authentication.

priv_lvl

This 8-bit field contains an enumerated value that identifies the privilege level requested by an authenticating user. For the current TACACS+ authentication implementation, this field always contains a value of 0x01 , indicating the user level.

authen-type

This 8-bit field contains an enumerated value that identifies the authentication methodology. Supported values are as follows:

0x01 ASCII — simple login, Oracle Enterprise Communications Broker prompts for username and password

0x02 PAP — as specified in RFC 1334

0x03 CHAP — as specified in RFC 1994

service

This 8-bit field contains an enumerated value that identifies the service requesting the authentication. For the current TACACS+ implementation, this field always contains a value of 0x01 , indicating user login authentication.

user_len

This 8-bit field contains the length of the user field in octets.

port_len

This 8-bit field contains the length of the port field in octets. As the port field is not used in the current TACACS+ authentication implementation, the port_len field always contains a value of 0 as specified in Section 4 of the TACACS+ draft RFC.

rem_addr_len

This 8-bit field contains the length of the rem_addr field in octets. As the rem_addr field is not used in the current TACACS+ authentication implementation, the rem_addr_len field always contains a value of 0 as specified in Section 4 of the TACACS+ draft RFC.

data_len

This 8-bit field contains the length of the data field in octets.

user

This variable length field contains the login name of the user to be authenticated.

port

This variable length field contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .

rem_addr

This variable length field contains the location of the user to be authenticated. This field contains the localhost address.

data

This optional variable length field contains miscellaneous data.

Authentication REPLY Packet

The TACACS+ daemon sends an authentication REPLY packet to the Oracle Enterprise Communications Broker in response to a authentication START or authentication CONTINUE packet. Depending on the contents of the status field, the authentication REPLY packet either ends the authentication transaction, or continues the transaction by requesting addition information needed by the authenticator.

The authentication REPLY packet format is as follows.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x1         |
+--------+--------+--------+--------+
|     (type field contains 0x1)     |
+--------+--------+-----------------+
| status |  flags |  server_msg_len |
|--------+--------+--------+--------+
|     data_len    |  server_msg ... |
+-----------------+-----------------+
|              data ...             |
+-----------------------------------+
```

status

This 16-bit field contains an enumerated value that specifies the current state of the authentication process. Supported values are as follows:

0x01 PASS — the user is authenticated, thus ending the session

0x02 FAIL — the user is rejected, thus ending the session

0x04 GETUSER — daemon request for the user name

0x05 GETPASS — daemon request for the user password

0x06 RESTART — restarts the transaction, possibly because the sequence number has wrapped, or possibly because the requested authentication type is not supported by the daemon

0x07 ERROR — reports an unrecoverable error

flags

This 8-bit field contains various flags that are not under user control.

server_msg_len

This 16-bit field contains the length of the server_msg field in octets. As the server_msg field is not used in REPLY packets sent by the current TACACS+ authentication implementation, the server_msg_len field always contains a value of 0 as specified in Section 4 of the TACACS+ draft RFC.

data_len

This 16-bit field contains the length of the data field in octets. As the data field is not used in REPLY packets sent by the current TACACS+ authentication implementation, the data_len field always contains a value of 0 as specified in Section 4 of the TACACS+ draft RFC.

server_msg

This optional variable length field contains a server message intended for display to the user. The current TACACS+ authentication implementation does not use this field.

data

This optional variable length field contains data pertinent to the authentication process. The current TACACS+ authentication implementation does not use this field.

Authentication CONTINUE Packet

The Oracle Enterprise Communications Broker, acting as a TACACS+ client, sends an authentication CONTINUE packet to the TACACS+ daemon in response to a REPLY message which requested additional data required by the authenticator.

The authentication CONTINUE packet format is as follows.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x1         |
+--------+--------+-----------------+
|   user_msg_len  |     data_len    |
|--------+--------+-----------------+
|  flags |       user_msg ...       |
+--------+--------------------------+
|              data ...             |
+-----------------------------------+
```

user_msg_len

This 16-bit field contains the length of the user_msg field in octets.

data_len

This 16-bit field contains the length of the data field in octets. As the data field is not used in the current TACACS+ authentication implementation, the data field always contains a value of 0 as specified in Section 4 of the TACACS+ draft RFC.

flags

This 8-bit field contains various flags that are not under user control.

user_msg

This variable length field contains a string that responds to an information request contained in a REPLY message.

data

This optional variable length field contains miscellaneous data, often in response to a daemon request. The current TACACS+ authentication implementation does not use the data field in Authentication CONTINUE packets.

Authentication Scenarios

Each of the supported user authentication scenarios is described in terms of packet flow in the following sections.

ASCII Authentication

The Oracle Enterprise Communications Broker initiates the authentication with an authentication START packet.

```
+-----------------------------------+
|           Common Header           |
|     minor_version contains 0x0    |
|         type contains 0x1         |
+--------+--------+--------+--------+
|action  |priv_lvl|authen_ |service |
|        |        |type    |        |
|  0x01  |  0x01  |  0x01  |  0x01  |
|--------+--------+--------+--------+
|user_len|port_len|rem_addr|data_len|
|        |        |_len    |        |
|    0   |    N   |    N   |    0   |
+--------+--------+--------+--------+
|                port               |
|               tty10               |
+-----------------------------------+
|              rem_addr             |
|         localhost address         |
+-----------------------------------+
```

- The action field specifies the requested authentication action — 0x01 for TAC_PLUSAUTHEN_LOGIN (authentication of a user login).
- The priv_lvl field specifies the privilege level requested by the user — 0x01 for TAC_PLUS_PRIV_LVL_USER.
- The authen_type field specifies the authentication methodology — 0x01 for TAC_PLUS_AUTHEN_TYPE_ASCII (simple login).
- The service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len and data_len fields contain a value of 0 , as required by the TACACS+ protocol.
- The port_len and rem_addr_len fields contain the length, in octets, of the port and rem_addr fields.
- The port field contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.

The TACACS+ daemon returns an authentication REPLY requesting the username.

```
+-----------------------------------+
|           Common Header           |
|     minor_version contains 0x0    |
|         type contains 0x1         |
+--------+--------+-----------------+
| status |  flags |  server_msg_len |
|  0x04  |        |        0        |
|--------+--------+-----------------+
|     data_len    |
|        0        |
+-----------------+
```

- The status field specifies a daemon request — 0x04 for TAC_PLUS_AUTH_STATUS_GETUSER (get username).
- The server_msg_len data_len fields both contain a value of 0 , as required by the TACACS+ protocol.

The Oracle Enterprise Communications Brokerresponds with an authentication CONTINUE packet.

```
+-----------------------------------+
|           Common Header           |
|     minor_version contains 0x0    |
|         type contains 0x1         |
+-----------------+-----------------+
|   user_msg_len  |     data_len    |
|                 |        0        |
|--------+--------+-----------------+
|  flags |       user_msg ...       |
+--------+--------------------------+
```

- The user_msg_len field contains the length, in octets, of the user_msg field.
- The data_len field contains a value of 0 , as required by the TACACS+ protocol.
- The user_msg field contains the username to be authenticated.

The TCACS+ daemon returns a second authentication REPLY requesting the user password.

```
+-----------------------------------+
|           Common Header           |
|     minor_version contains 0x0    |
|         type contains 0x1         |
+--------+--------+--------+--------+
| status |  flags |  server_msg_len |
|  0x05  |        |        0        |
|--------+--------+--------+--------+
|     data_len    |
|        0        |
+-----------------+
```

- The status field specifies a daemon request — 0x05 for TAC_PLUS_AUTH_STATUS_GETPASS (get user password).
- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.

The Oracle Enterprise Communications Broker responds with a second authentication CONTINUE packet.

```
+-----------------------------------+
|           Common Header           |
|     minor_version contains 0x0    |
|         type contains 0x1         |
+-----------------+-----------------+
|   user_msg_len  |     data_len    |
|                 |        0        |
|--------+--------+--------+--------+
|  flags |       user_msg ...       |
+--------+--------------------------+
```

- The user_msg_len field contains the length, in octets, of the user_msg field.
- The data_len field contains a value of 0 , as required by the TACACS+ protocol.
- The user_msg field contains the user password to be authenticated.
- Other, optional fields are not used.

The TACACS+ daemon returns a third authentication REPLY reporting the authentication result, and terminating the authentication session.

```
+------------------------------------+
|           Common Header            |
|     minor_version contains 0x0     |
|         type contains 0x1          |
+--------+--------+------------------+
| status |  flags |  server_msg_len  |
|  0x01  |        |        0         |
|--------+--------+------------------+
|     data_len    |
|        0        |
+-----------------+
```

- The status field specifies the authentication result — 0x01 for TAC_PLUS_AUTH_STATUS_PASS (authorization succeeds), or 0x02 for TAC_PLUS_AUTH_STATUS_FAIL (authorization fails).
- The server_msg_len , and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.

PAP Authentication

The Oracle Enterprise Communications Broker initiates the authentication with an authentication START packet.

```
+-----------------------------------+
|           Common Header           |
|     minor_version contains 0x1    |
|         type contains 0x1         |
+--------+--------+--------+--------+
|action  |priv_lvl|authen_ |service |
|        |        |type    |        |
|  0x01  |  0x01  |  0x02  |  0x01  |
|--------+--------+--------+--------+
|user_len|port_len|rem_addr|data_len|
|        |        |_len    |        |
|    N   |    N   |    N   |    N   |
+--------+--------+--------+--------+
|               user                |
+-----------------------------------+
|               port                |
|              tty10                |
+-----------------------------------+
|              rem_addr             |
|         localhost address         |
+-----------------------------------+
|              data ...             |
+-----------------------------------+
```

- The action field specifies the requested authentication action — 0x01 for TAC_PLUSAUTHEN_LOGIN (authentication of a user login).
- The priv_lvl field specifies the privilege level requested by the user — 0x01 for TAC_PLUS_PRIV_LVL_USER.
- The authen_type field specifies the authentication methodology — 0x02 for TAC_PLUS_AUTHEN_TYPE_PAP (PAP login).
- The service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len field contains the length, in octets, of the user field.
- The port_len field contains the length, in octets, of the port field.
- The rem_addr_len field contains the length, in octets, of the rem_addr field.
- The data_len field contains the length, in octets, of the date field.
- The user field contains the username to be authenticated.
- The port field contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .

- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.
- The data field contains the password to be authenticated.

The TCACS+ daemon returns an authentication REPLY reporting the authentication result.

```
+-----------------------------------+
|           Common Header           |
|     minor_version contains 0x1    |
|         type contains 0x1         |
+--------+--------+-----------------+
| status |  flags |  server_msg_len |
|  0x01  |        |        0        |
|--------+--------+-----------------+
|     data_len    |
|        0        |
+-----------------+
```

- The status field specifies the authentication result — 0x01 for TAC_PLUS_AUTH_STATUS_PASS (authorization succeeds), or 0x02 for TAC_PLUS_AUTH_STATUS_FAIL (authorization fails).
- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.
- Other, optional fields are not used.

CHAP Authentication

The Oracle Enterprise Communications Broker initiates the authentication with an authentication START packet.

```
+-----------------------------------+
|           Common Header           |
|     minor_version contains 0x1    |
|         type contains 0x1         |
+--------+--------+--------+--------+
|action  |priv_lvl|authen_ |service |
|        |        |type    |        |
|  0x01  |  0x01  |  0x03  |  0x01  |
|--------+--------+--------+--------+
|user_len|port_len|rem_addr|data_len|
|        |        |_len    |        |
|    N   |    N   |    N   |    N   |
+--------+--------+--------+--------+
|                user               |
+-----------------------------------+
|                port               |
|               tty10               |
+-----------------------------------+
|              rem_addr             |
|         localhost address         |
+-----------------------------------+
|              data ...             |
+-----------------------------------+
```

- The action field specifies the requested authentication action — 0x01 for TAC_PLUSAUTHEN_LOGIN (authentication of a user login).
- The priv_lvl field specifies the privilege level requested by the user — 0x01 for TAC_PLUS_PRIV_LVL_USER.
- The authen_type field specifies the authentication methodology — 0x03 for TAC_PLUS_AUTHEN_TYPE_CHAP (CHAP login).
- The service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len field contains the length, in octets, of the user field.
- The port_len field contains the length, in octets, of the port field.

- The rem_addr_len field contains the length, in octets, of the rem_addr field.
- The data_len field contains the length, in octets, of the date field.
- The user field contains the username to be authenticated.
- The port field contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.
- The data field contains the password to be authenticated.

The TCACS+ daemon returns an authentication REPLY reporting the authentication result.

```
+----------------------------------+
|           Common Header          |
|     minor_version contains 0x1   |
|         type contains 0x1        |
+--------+--------+----------------+
| status |  flags |  server_msg_len |
|  0x01  |        |        0       |
|--------+--------+----------------+
|     data_len    |
|        0        |
+-----------------+
```

- The status field specifies the authentication result — 0x01 for TAC_PLUS_AUTH_STATUS_PASS (authorization succeeds), or 0x02 for TAC_PLUS_AUTH_STATUS_FAIL (authorization fails).
- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.
- Other, optional fields are not used.

TACACS+ Authorization

The Oracle Enterprise Communications Broker uses TACACS+ services to provide administrative authorization. With TACACS+ authorization enabled, each individual ACLI command issued by an admin user is authorized by the TACACS+ authorization service. The Oracle Enterprise Communications Broker replicates each ACLI command in its entirety, sends the command string to the authorization service, and suspends command execution until it receives an authorization response. If TACACS+ grants authorization, the pending command is executed; if authorization is not granted, the Oracle Enterprise Communications Broker does not execute the ACLI command, and displays an appropriate error message.

The daemon’s authorization decisions are based on a database lookup. Data base records use regular expressions to associate specific command string with specific users. The construction of such records is beyond the scope of this document.

Authorization Message Exchange

All TACACS+ authorization packets consist of a common header and a message body. Authorization packets are of two types: REQUEST and RESPONSE.

The REQUEST packet, which initiates an authorization session, is always sent by the Oracle Enterprise Communications Broker. Upon receipt of every REQUEST, the daemon must answer with a RESPONSE packet. In the current TACACS+ implementation, the RESPONSE packet must contain an authorization decision (pass or fail). The exchange of a single REQUEST and the corresponding RESPONSE completes the authorization session.

Authorization REQUEST Packet

The Oracle Enterprise Communications Broker, acting as a TACACS+ client, sends an authorization REQUEST packet to the TACACS+ daemon to initiate an authorization session.

The authorization REQUEST packet format is as follows.

```
+----------------------------------+
|           Common Header          |
|                                   |
|          type contains 0x2        |
+--------+--------+--------+--------+
|authen_ |priv_lvl|authen_ |authen- |
|method  |        |type    |service |
|--------+--------+--------+--------+
|user_len|port_len|rem_addr|arg_cnt |
|        |        |_len    |        |
+----+---+--------+--------+--------+
|arg1_len|arg2_len|  ...   |argN_len|
|        |        |        |        |
+--------+--------+--------+--------+
|              user ...             |
+-----------------------------------+
|              port ...             |
+-----------------------------------+
|            rem-addr ...           |
+-----------------------------------+
|              arg1 ...             |
+-----------------------------------+
|              arg2 ...             |
+-----------------------------------+
|              argN ...             |
+-----------------------------------+
```

authen_method

This 8-bit field contains an enumerated value that identifies the method used to authenticate the authorization subject — that is, an admin user. Because the admin user was authenticated locally by the Oracle Enterprise Communications Broker, this field always contains a value of 0x05 , indicating authentication by the requesting client.

priv_lvl

This 8-bit field contains an enumerated value that identifies the privilege level associated with the authorization subject. For the current TACACS+ authorization implementation, this field always contains a value of 0x00 .

authen-type

This 8-bit field contains an enumerated value that identifies the methodology. used to authenticate the authorization subject. Because the admin user was authenticated with a simple username/password exchange, this field always contains a value of 0x01 , indicating ascii login.

authen_service

This 8-bit field contains an enumerated value that identifies the service that requested authentication. Because an admin user is authenticated with a simple username/password exchange, this field always contains a value of 0x01 , the login service.

user_len

This 8-bit field contains an integer that specifies the length, in octets, of the user field.

port_len

This 8-bit field contains an integer that specifies the length, in octets, of the port field.

rem_addr_len

This 8-bit field contains an integer that specifies the length, in octets, of the rem_addr field.

arg_cnt

This 8-bit field contains an integer that specifies the number or arguments contained with the REQUEST. Given the design of the current TACACS+ implementation, this field always contains a value of 0x02 .

arg1_len

This 8-bit field contains an integer that specifies the length, in octets, of the first argument.

Subsequent fields contain the length of each sequential argument.

user

This variable length field contains the login name of the user to be authorized.

port

This variable length field contains the name of the Oracle Enterprise Communications Broker port on which authorization is taking place. Following Cisco Systems convention, this field contains the string tty10 .

rem_addr

This variable length contains the location of the user to be authorized. This field contains the localhost address.

arg...

This variable length field contains a TACACS+ attribute value pair (AVP); each arg field holds a single AVP.

A TACACS+ AVP is an ASCII string with a maximum length of 255 octets. The string consists of the attribute name and its assigned value separated by either an equal sign (=) or by an asterisk (*). The equal sign (=) identifies a mandatory argument, one that must be understood and processed by the TACACS+ daemon; the asterisk (*) identifies an optional argument that may be disregarded by either the client or daemon.

Administrative authorization requires the use of only two TACACS+ AVPs: service and cmd .

The service AVP identifies the function to be authorized. In the case of the current implementation, the attribute value is always shell . Consequently the attribute takes the follow format:

service=shell

The cmd AVP identifies the specific ACLI command to be authorized. The command is passed in its entirety, from the administrative configuration root, **`configure terminal`**, through the final command argument. For example,

cmd=configure terminal security authentication type tacacsplus

Note the equal sign (=) used in the attribute examples, indicating that both are mandatory arguments.

Authorization RESPONSE Packet

The TACACS+ daemon sends an authorization RESPONSE packet to the Oracle Enterprise Communications Broker to report authorization results.

The authorization RESPONSE packet format is as follows.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x2         |
+--------+--------+-----------------+
|status  |arg_cnt | server_msg len  |
|        |        |                 |
|--------+--------+--------+--------+
|     data_len    |arg1_len|arg2_len|
|                 |        |        |
+--------+--------+--------+--------+
|   ...  |argN_len|    server_msg   |
|        |        |                 |
+--------+--------+-----------------+
|              data ...             |
+-----------------------------------+
|              arg1 ...             |
+-----------------------------------+
|              arg2 ...             |
+-----------------------------------+
|              argN ...             |
+-----------------------------------+
```

status

This 8-bit field contains an enumerated value that specifies the results of the authorization process. Supported values are 0x01 (Pass), 0x10 (Fail), and 0x11 (Error). Fail indicates that the authorization service rejected the proposed operation, while Error indicates the authorization service failed

If authorization succeeds (status=0x01), the ACLI command is executed; if authorization fails, for whatever the reason (status=0x10 or 0x11), the ACLI command is not executed, and an appropriate error message is generated.

arg_cnt

This 8-bit field contains an integer that specifies the number or arguments contained with the RESPONSE. Given the design of the current TACACS+ implementation, this field always contains a value of 0x02 .

server_msg_len

This 16-bit field contains an integer that specifies the length, in octets, of the server_msg field.

data_len

This 16-bit field contains an integer that specifies the length, in octets, of the data field.

arg1_len

This 8-bit field contains an integer that specifies the length, in octets, of the first argument.

Subsequent fields contain the length of each sequential argument.

server-msg

This optional variable length field contains a string that can be presented to the user.

data

This optional variable length field contains a string that can be presented to an administrative display, console, or log.

arg...

This optional variable length field contains a TACACS+ attribute value pair (AVP); each arg field holds a single AVP.

No arguments are generated in RESPONSE packets within the current TACACS+ implementation.

Authorization Pass

The Oracle Enterprise Communications Broker initiates the authorization with an authorization REQUEST packet.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x2         |
+--------+--------+--------+--------+
|authen_ |priv_lvl|authen_ |authen_ |
|method  |        |type    |service |
|  0x05  |  0x00  |  0x01  |  0x01  |
|--------+--------+--------+--------+
|user_len|port_len|rem_addr|arg_cnt |
|        |        |_len    |        |
|    N   |    N   |    N   |    2   |
+--------+--------+--------+--------+
|arg1_len|arg2_len|      user ...   |
|        |        |                 |
|    N   |    N   |    login name   |
+--------+--------+-----------------+
|                port               |
|               tty10               |
+-----------------------------------+
|              rem_addr             |
|         localhost address         |
+-----------------------------------+
|                arg1               |
|                AVP                |
|           service=shell           |
+-----------------------------------+
|                arg2               |
|                AVP                |
|  cmd=configure terminal security  |
+-----------------------------------+
```

- The authen_method field specifies the method used to authenticate the subject — 0x05 for TAC_PLUS_AUTHEN_METHOD_LOCAL (authentication by the client).
- The priv_lvl field specifies the privilege level requested by the user — 0x00 for TAC_PLUS_PRIV_LVL_MIN.
- The authen_type field specifies the authentication methodology — 0x01 for TAC_PLUS_AUTHEN_TYPE_ASCII (simple login).
- The authen_ service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len field contains the length, in octets, of the user field.
- The port_len field contains the length, in octets, of the port field.
- The rem_addr_len field contains the length, in octets, of the rem_addr field.
- The arg_cnt field contains the number of arguments in the message body.
- The arg1_len field contains the length, in octets, of the service AVP.
- The arg2_len field contains the length, in octets, of the service AVP.
- The user field contains the login name of an admin user.
- The port field contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.
- The arg1 field contains the mandatory service AVP.
- The arg2 field contains the mandatory cmd AVP.

The TACACS+ daemon returns a authorization RESPONSE reporting the status, and terminating the authorization session.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x2         |
+--------+--------+-----------------+
| status |arg_cnt |  server_msg_len |
|  0x01  |   0    |        0        |
|--------+--------+-----------------+
|     data_len    |
|        0        |
+-----------------+
```

- The status field specifies the authorization status — 0x01 for TAC_PLUS_AUTHOR_STATUS_PASS_ADD (authorization approved).
- The arg_cnt field contains a value of 0 — the authorization RESPONSE returns no arguments.
- The server_msg_len and data_len fields both contain a value of 0, as required by the TACACS+ protocol.

Authorization Fail

The Oracle Enterprise Communications Broker initiates the authorization with an authorization REQUEST packet.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x2         |
+--------+--------+--------+--------+
|authen_ |priv_lvl|authen_ |authen_ |
|method  |        |type    |service |
|  0x05  |  0x00  |  0x01  |  0x01  |
|--------+--------+--------+--------+
|user_len|port_len|rem_addr|arg_cnt |
|        |        |_len    |        |
|    N   |    N   |    N   |    2   |
+--------+--------+--------+--------+
|arg1_len|arg2_len|      user ...   |
|        |        |                 |
|    N   |    N   |    login name   |
+--------+--------+-----------------+
|                port               |
|               tty10               |
+-----------------------------------+
|              rem_addr             |
|         localhost address         |
+-----------------------------------+
|                arg1               |
|                AVP                |
|            service=shell          |
+-----------------------------------+
|                arg2               |
|                AVP                |
|   cmd=configure terminal scurity  |
+-----------------------------------+
```

- The authen_method field specifies the method used to authenticate the administrative subject — 0x05 for TAC_PLUS_AUTHEN_METHOD_LOCAL (authentication by the client).
- The priv_lvl field specifies the privilege level requested by the user — 0x00 for TAC_PLUS_PRIV_LVL_MIN.
- The authen_type field specifies the authentication methodology — 0x01 for TAC_PLUS_AUTHEN_TYPE_ASCII (simple login).
- The authen_ service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len field contains the length, in octets, of the user field.
- The port_len field contains the length, in octets, of the port field.
- The rem_addr_len field contains the length, in octets, of the rem-addr field.
- The arg_cnt field contains the number of arguments in the message body.
- The arg1_len field contains the length, in octets, of the service AVP.
- The arg2_len field contains the length, in octets, of the service AVP.
- The user field contains the login name of an admin user.
- The port field contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .

- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.
- The arg1 field contains the mandatory service AVP.
- The arg2 field contains the mandatory cmd AVP.

The TACACS+ daemon returns an authorization RESPONSE reporting the status, and terminating the authorization session.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x2         |
+--------+--------+--------+--------+
| status |arg_cnt |  server_msg_len |
|  0x10  |   0    |        0        |
|--------+--------+--------+--------+
|     data_len    |
|        0        |
+-----------------+
```

- The status field specifies the authorization status — 0x10 for TAC_PLUS_AUTHOR_STATUS_FAIL (authorization rejected).
- The arg_cnt field contains a value of 0 — the authorization RESPONSE returns no arguments.
- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.

TACACS+ Accounting

The Oracle Enterprise Communications Broker uses TACACS+ accounting to log administrative actions. With accounting enabled, each individual ACLI command executed by an admin user is logged by the accounting service.

Accounting Message Exchange

All TACACS+ accounting packets consist of a common header and a message body. Accounting packets are of two types: REQUEST and REPLY.

The REQUEST packet has three variant forms. The START variant initiates an accounting session; the STOP variant terminates an accounting session; the WATCHDOG variant updates the current accounting session. REQUEST packets are always sent by the Oracle Enterprise Communications Broker. Upon receipt of every REQUEST, the daemon must answer with a REPLY packet.

A TACACS+ accounting session proceeds as follows.

1. Immediately following successful authorization of an admin user, the Oracle Enterprise Communications Broker sends an accounting REQUEST START packet.
2. The daemon responds with an accounting REPLY packet, indicating that accounting has started.
3. For each ACLI command executed by an admin user, the Oracle Enterprise Communications Broker sends an accounting REQUEST WATCHDOG packet requesting accounting of the ACLI command. As the Oracle Enterprise Communications Broker sends the WATCHDOG only after an admin user's access to the ACLI command is authorized, the accounting function records only those commands executed by the user, not those commands for which authorization was not granted.
4. The daemon responds with an accounting REPLY packet, indicating that the ACLI operation has been recorded by the accounting function.
5. Steps 3 and 4 are repeated for each authorized ACLI operation.
6. Immediately following logout (or timeout) of an admin user, the Oracle Enterprise Communications Broker sends an accounting REQUEST STOP packet.
7. The daemon responds with an accounting REPLY packet, indicating that accounting has stopped.

Accounting REQUEST Packet

The Oracle Enterprise Communications Broker, acting as a TACACS+ client, sends an accounting REQUEST START variant to the TACACS+ daemon following the successful authorization of an admin user. It sends an accounting REQUEST WATCHDOG variant to the daemon following the authorization of an admin user’s access to an ACLI command. It sends an accounting REQUEST STOP variant to the daemon at the conclusion of the ACLI session.

The accounting REQUEST packet format is as follows.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x3         |
+--------+--------+--------+--------+
| flags  |authen_ |priv_lvl|authen- |
|        |method  |        |type    |
|--------+--------+--------+--------+
|authen_ |user_len|port_len|rem_addr|
|service |        |        |_len    |
+----+---+--------+--------+--------+
|arg_cnt |arg1_len|arg2_len|argN_len|
|        |        |        |        |
+--------+--------+--------+--------+
|argN_len|         user ...         |
+--------+--------------------------+
|              port ...             |
+-----------------------------------+
|            rem-addr ...           |
+-----------------------------------+
|              arg1 ...             |
+-----------------------------------+
|              arg2 ...             |
+-----------------------------------+
|              argN ...             |
+-----------------------------------+
```

flags

This 8-bit field contains an enumerated value that identifies the accounting REQUEST variant.

0x2 — START

0x4 — STOP

0x8 — WATCHDOG

authen_method

This 8-bit field contains an enumerated value that identifies the method used to authenticate the accounting subject — that is, an admin user. Because an admin user is authenticated locally by the Oracle Enterprise Communications Broker, this field always contains a value of 0x05 , indicating authentication by the requesting client.

priv_lvl

This 8-bit field contains an enumerated value that identifies the privilege level associated with the accounting subject. For the current TACACS+ accounting implementation, this field always contains a value of 0x00 .

authen-type

This 8-bit field contains an enumerated value that identifies the methodology. used to authenticate the accounting subject. Because an admin user is authenticated with a simple username/password exchange, this field always contains a value of 0x01 , indicating ascii login.

authen_service

This 8-bit field contains an enumerated value that identifies the service that requested authentication. Because an admin user is authenticated with a simple username/password exchange, this field always contains a value of 0x01 , the login service.

user_len

This 8-bit field contains an integer that specifies the length, in octets, of the user field.

port_len

This 8-bit field contains an integer that specifies the length, in octets, of the port field.

rem_addr_len

This 8-bit field contains an integer that specifies the length, in octets, of the rem_addr field.

arg_cnt

This 8-bit field contains an integer that specifies the number or arguments contained with the accounting REQUEST.

arg1_len

This 8-bit field contains an integer that specifies the length, in octets, of the first argument.

Subsequent fields contain the length of each sequential argument.

user

This variable length field contains the login name of the accounting subject.

port

This variable length field contains the name of the Oracle Enterprise Communications Broker port on accounting is taking place. Following Cisco System convention, this field always contains the string tty10 .

rem_addr

This variable length contains the location of the authorization subject. This field always contains the localhost address.

arg...

This variable length field contains a TACACS+ attribute value pair (AVP); each arg field holds a single AVP.

A TACACS+ AVP is an ASCII string with a maximum length of 255 octets. The string consists of the attribute name and its assigned value separated by either an equal sign (=) or by an asterisk (*). The equal sign (=) identifies a mandatory argument, one that must be understood and processed by the TACACS+ daemon; the asterisk (*) identifies an optional argument that may be disregarded by either the client or daemon.

Administrative accounting requires the use of five TACACS+ AVPs: service, task-id, start_time, and stop_time.

The task_id AVP, included in accounting REQUEST START, STOP, and WATCHDOG variants, correlates session initiation, watchdog updates, and termination packets; each associated START, STOP, and WATCHDOG packet must contain matching task-id AVPs.

task_id=13578642

The start_time AVP, included in accounting REQUEST START and WATCHDOG variants, specifies the time at which a specific accounting request was initiated. The start time is expressed as the number of seconds elapsed since January 1, 1970 00:00:00 UTC.

start_time=1286790650

The stop_time AVP, included in accounting REQUEST STOP variants, specifies the time at which a specific accounting session was terminated. The stop time is expressed as the number of seconds elapsed since January 1, 1970 00:00:00 UTC.

stop_time=1286794250

The service AVP, included in accounting REQUEST START, STOP, and WATCHDOG variants, identifies the function subject to accounting. In the case of the current implementation, the attribute value is always shell . Consequently the attribute takes the follow format:

service=shell

The cmd AVP, included in accounting REQUEST WATCHDOG variants, identifies the specific ACLI command to be processed by the accounting service. The command is passed in its entirety, from the administrative configuration root, **`configure terminal`**, through the final command argument. For example,

cmd=configure terminal security authentication type tacacsplus

Note the equal sign (=) used in the attribute examples, indicating that all are mandatory arguments.

Accounting REPLY Packet

The TACACS+ daemon sends an accounting REPLY packet to the Oracle Enterprise Communications Broker to report accounting results.

The accounting REPLY packet format is as follows.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x3         |
+-----------------+--------+--------+
|  server_msg_len |     data_len    |
|--------+--------+-----------------+
| status |      server_msg ...      |
+--------+--------------------------+
|              data ...             |
+-----------------------------------+
```

server_msg_len

This 16-bit field contains the length, in octets, of the server_msg field.

data_len

This 16-bit field contains the length, in octets, of the data field.

status

This 8-bit field contains the status of the previous accounting request. Supported values are:

0x1 — Success

0x2 — Error/Failure

server_msg

This optional variable length field can contain a message intended for display to the user. This field is unused in the current TACACS+ implementation.

data

This optional variable length field can contain miscellaneous data. This field is unused in the current TACACS+ implementation.

Accounting Scenario

The Oracle Enterprise Communications Broker initiates the accounting session with an accounting REQUEST START.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x3         |
+--------+--------+--------+--------+
| flags  |authen_ |priv_lvl|authen- |
|        |method  |        |type    |
|  0x02  |  0x05  |  0x00  |  0x01  |
|--------+--------+--------+--------+
|authen_ |user_len|port_len|rem_addr|
|service |        |        |_len    |
|  0X01  |    N   |    N   |    N   |
+----+---+--------+--------+--------+
|arg_cnt |arg1_len|arg2_len|arg3_len|
|    3   |    N   |    N   |    N   |
+--------+--------+--------+--------+
|                user               |
|    login name of an admin user    |
+-----------------------------------+
|                port               |
|               tty10               |
+-----------------------------------+
|              rem_addr             |
|         localhost address         |
+-----------------------------------+
|                AVP                |
|          task-id=13578642         |
+-----------------------------------+
|                AVP                |
|       start_time=1286790650       |
+-----------------------------------+
|                AVP                |
|           service=shell           |
+-----------------------------------+
```

- The flags field contains an enumerated value (0x02) that identifies an accounting REQUEST START.
- The authen_method field specifies the method used to authenticate the ACCOUNTING subject — 0x05 for TAC_PLUS_AUTHEN_METHOD_LOCAL (authentication by the client).
- The priv_lvl field specifies the privilege level requested by the user — 0x00 for TAC_PLUS_PRIV_LVL_MIN.
- The authen_type field specifies the authentication methodology — 0x01 for TAC_PLUS_AUTHEN_TYPE_ASCII (simple login).
- The authen_ service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len field contains the length, in octets, of the user field.
- The port_len field contains the length, in octets, of the port field.
- The rem_addr_len field contains the length, in octets, of the rem_addr field.
- The arg_cnt field contains the number of arguments in the message body.
- The arg1_len field contains the length, in octets, of the task_id AVP.
- The arg2_len field contains the length, in octets, of the start_time AVP.
- The arg3_len field contains the length, in octets, of the service AVP.
- The user field contains the login name of an admin user.
- The port field contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .

- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.
- The arg1 field contains the mandatory task_id AVP.
- The arg2 field contains the mandatory start_time AVP.
- The arg3 field contains the mandatory service AVP.

The TACACS+ daemon returns an accounting REPLY reporting the status, indicating that accounting has started.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x3         |
+-----------------+-----------------+
|  server_msg_len |     data_len    |
|        0        |        0        |
|--------+--------+-----------------+
| status |
|  0x01  |
+--------+
```

- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.
- The status field specifies the authorization status — 0x01 for TAC_PLUS_ACCT_STATUS_SUCCESS (accounting processed).

The Oracle Enterprise Communications Broker reports ACLI command execution with an accounting REQUEST WATCHDOG.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x3         |
+--------+--------+--------+--------+
| flags  |authen_ |priv_lvl|authen- |
|        |method  |        |type    |
|  0x08  |  0x05  |  0x00  |  0x01  |
|--------+--------+--------+--------+
|authen_ |user_len|port_len|rem_addr|
|service |        |        |_len    |
|  0X01  |    N   |    N   |    N   |
+----+---+--------+--------+--------+
|arg_cnt |arg1_len|arg2_len|arg3_len|
|    4   |    N   |    N   |    N   |
+--------+--------+--------+--------+
|arg4_len|           user           |
|        | login name of admin user |
+--------+--------------------------+
|                port               |
|               tty10               |
+-----------------------------------+
|              rem_addr             |
|         localhost address         |
+-----------------------------------+
|                AVP                |
|          task-id=13578642         |
+-----------------------------------+
|                AVP                |
|       start_time=1286790650       |
+-----------------------------------+
|                AVP                |
|           service=shell           |
+-----------------------------------+
|               AVP                 |
|  cmd=configure terminal security  |
+-----------------------------------+
```

- The flags field contains an enumerated value (0x08) that identifies an accounting REQUEST WATCHDOG.
- The authen_method field specifies the method used to authenticate the ACCOUNTING subject — 0x05 for TAC_PLUS_AUTHEN_METHOD_LOCAL (authentication by the client).
- The priv_lvl field specifies the privilege level requested by the user — 0x00 for TAC_PLUS_PRIV_LVL_MIN.
- The authen_type field specifies the authentication methodology — 0x01 for TAC_PLUS_AUTHEN_TYPE_ASCII (simple login).
- The authen_ service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len field contains the length, in octets, of the user field.
- The port_len field contains the length, in octets, of the port field.
- The rem_addr_len field contains the length, in octets, of the rem_addr field.
- The arg_cnt field contains the number of arguments in the message body.
- The arg1_len field contains the length, in octets, of the task_id AVP.
- The arg2_len field contains the length, in octets, of the start_time AVP.
- The arg3_len field contains the length, in octets, of the service AVP.
- The arg4_len field contains the length, in octets, of the cmd AVP.
- The user field contains the login name of an admin user.
- The port field contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.
- The arg1 field contains the mandatory task_id AVP.
- The arg2 field contains the mandatory start_time AVP.
- The arg3 field contains the mandatory service AVP.
- The arg4 field contains the mandatory cmd AVP.

The TACACS+ daemon returns an accounting REPLY reporting the status, indicating that the ACLI operation has been processed.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x3         |
+-----------------+-----------------+
|  server_msg_len |     data_len    |
|        0        |        0        |
|--------+--------+-----------------+
| status |
|  0x01  |
+--------+
```

- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.
- The status field specifies the authorization status — 0x01 for TAC_PLUS_ACCT_STATUS_SUCCESS (accounting processed).

The Oracle Enterprise Communications Broker reports an admin user logout or timeout with an accounting REQUEST STOP.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x3         |
+--------+--------+--------+--------+
| flags  |authen_ |priv_lvl|authen- |
|        |method  |        |type    |
|  0x04  |  0x05  |  0x00  |  0x01  |
|--------+--------+--------+--------+
|authen_ |user_len|port_len|rem_addr|
|service |        |        |_len    |
|  0X01  |    N   |    N   |    N   |
+----+---+--------+--------+--------+
|arg_cnt |arg1_len|arg2_len|arg3_len|
|    3   |    N   |    N   |    N   |
+--------+--------+--------+--------+
|                user               |
|    login name of an admin user    |
+-----------------------------------+
|                port               |
|               tty10               |
+-----------------------------------+
|              rem_addr             |
|         localhost address         |
+-----------------------------------+
|                AVP                |
|          task-id=13578642         |
+-----------------------------------+
|                AVP                |
|        stop_time=1286790650       |
+-----------------------------------+
|                AVP                |
|           service=shell           |
+-----------------------------------+
```

- The flags field contains an enumerated value (0x04) that identifies an accounting REQUEST STOP.
- The authen_method field specifies the method used to authenticate the ACCOUNTING subject — 0x05 for TAC_PLUS_AUTHEN_METHOD_LOCAL (authentication by the client).
- The priv_lvl field specifies the privilege level requested by the user — 0x00 for TAC_PLUS_PRIV_LVL_MIN.
- The authen_type field specifies the authentication methodology — 0x01 for TAC_PLUS_AUTHEN_TYPE_ASCII (simple login).
- The authen_ service field specifies the requesting service — 0x01 for TAC_PLUS_AUTHEN_SVC_LOGIN (login service).
- The user_len field contains the length, in octets, of the user field.
- The port_len field contains the length, in octets, of the port field.
- The rem_addr_len field contains the length, in octets, of the rem_addr field.
- The arg_cnt field contains the number of arguments in the message body.
- The arg1_len field contains the length, in octets, of the task_id AVP.
- The arg2_len field contains the length, in octets, of the start_time AVP.
- The arg3_len field contains the length, in octets, of the service AVP.
- The user field contains the login name of an admin user.
- The port field contains the name of the Oracle Enterprise Communications Broker port on which authentication is taking place. Following Cisco Systems convention, this field contains the string tty10 .
- The rem_addr field specifies the location of the user to be authenticated. This field contains the localhost address.
- The arg1 field contains the mandatory task_id AVP.
- The arg2 field contains the mandatory start_time AVP.
- The arg3 field contains the mandatory service AVP.

The TACACS+ daemon returns an accounting REPLY reporting the status, indicating that accounting has terminated.

```
+-----------------------------------+
|           Common Header           |
|                                   |
|         type contains 0x3         |
+-----------------+-----------------+
|  server_msg_len |     data_len    |
|        0        |        0        |
|--------+--------+-----------------+
| status |
|  0x01  |
+--------+
```

- The server_msg_len and data_len fields both contain a value of 0 , as required by the TACACS+ protocol.
- The status field specifies the authorization status — 0x01 for TAC_PLUS_ACCT_STATUS_SUCCESS (accounting processed).

Managing TACACS+ Operations

TACACS+ management is supported by the following utilities.

TACACS+ MIB

An Oracle proprietary MIB provides external access to TACACS+ statistics.

MIB counters are contained in the apSecurityTacacsPlusStatsTable that is defined as follows.

```
SEQUENCE {
    apSecurityTacacsPlusCliCommands                Counter32
    apSecurityTacacsPlusSuccess Authentications    Counter32
    apSecurityTacacsPlusFailureAuthentications     Counter32
    apSecurityTacacsPlusSuccess Authorizations     Counter32
    apSecurityTacacsPlusFailureAuthorizations      Counter32
}
```

apSecuritysTacacsPlusStats Table (1.3.6.1.4.1.9148.3.9.9.4)		
Object Name	**Object OID**	**Description**
apSecurityTacacsCliCommands	1.3.6.1.4.1.9148.3.9.1.4.3	Global counter for ACLI commands sent to TACACS+ Accounting
apSecurityTacacsSuccess Authentications	1.3.6.1.4.1.9148.3.9.1.4.4	Global counter for the number of successful TACACS+ authentications
apSecurityTacacsFailureAuthentications	1.3.6.1.4.1.9148.3.9.1.4.5	Global counter for the number of unsuccessful TACACS+ authentications
apSecurityTacacsSuccess Authorizations	1.3.6.1.4.1.9148.3.9.1.4.6	Global counter for the number of successful TACACS+ authorizations
apSecurityTacacsFailure Authorizations	1.3.6.1.4.1.9148.3.9.1.4.7	Global counter for the number of unsuccessful TACACS+ authorizations

SNMP Trap

SNMP traps are issued when

- a TACACS+ daemon becomes unreachable
- an unreachable TACACS+ daemon becomes reachable

- an authentication error occurs
- an authorization error occurs

TACACS+ Faults

The Oracle Enterprise Communications Broker supports two TACACS+ traps, apSysMgmtTacacsDownTrap and apSysMgmtTacacsDownClearTrap.

The apSysMgmtTacacsDownTrap is generated when a TACACS+ server becomes unreachable.

The apSysMgmtTacacsDownClearTrap is generated when a TACACS+ server that was unreachable becomes reachable.

The ECB searches for a TACACS+ server until it finds an available one and then stops searching. However, in the TACACS+ SNMP implementation, SNMP expects the ECB to make connection attempts to all servers. When there is only one TACACS+ server and that server goes down, the ECB behaves normally, sending a apSysMgmtTacacsDownTrap trap when the server goes down, and a apSysMgmtTacacsDownClearTrap trap when the server comes back up. When there is more than one TACACS+ server and the active server goes down, an apSysMgmtTacacsDownTrap trap is sent, indicating that some servers are down and the next server is tried. If all servers fail, an apSysMgmtTacacsDownTrap is sent indicating that all servers are down. If one of the servers comes back up while the rest are still down, an apSysMgmtTacacsDownTrap is sent indicating that some servers are still down.

TACACS+ Logging

All messages between the Oracle Enterprise Communications Broker and the TACACS+ daemon are logged in a cleartext format, allowing an admin user to view all data exchange, except for password information.

TACACS+ Configuration

Configuration of TACACS+ consists of the following steps.

1. Enable TACACS+ client services
2. Specify one or more TACACS+ servers (daemons)

Add TACACS+ Authentication and Servers

To configure TACACS+, you enable TACACS+ client services and specify one or more TACACS+ servers.

1. Access the Login Authentication configuration object.

   Configuration > **Security** > **Login Authentication**.
2. On the Modify Authentication page, do the following:

Attributes	Instructions
Source port	Range: 1645-1812. Default: 1812.
Type	Select TACACS from the drop-down list.
Protocol	Select acsii for the authentication protocol.
TACACS accounting	Select to enable accounting of admin operations. Default: enabled.
Server assigned privilege	Select to allow only Admin users to use configuration commands. Default: Disabled.
Allow local authentication	Select to enable local authentication. Default: Disabled.
Login as Admin	Select to enable logging in as Admin.
Management strategy	Select an authentication management strategy from the drop-down list.

Attributes	Instructions
	• Use either Hunt or Round-Robin when using multiple TACACS+ servers. • Use Hunt when using a single TACACS+ server. Default: Hunt.
Management servers	Click **Add**, and do the following to add one or more authentication management servers: **1.** Enter the IP address of a management server. **2.** (Optional) Click **Apply / Add Another**. **3.** **OK**.
TACACS servers	Click **Add**, and do the following: **1.** Address—Enter the IP address of this server. **2.** Port—Enter the port number of the server you want to receive TACACS+ client requests. Range: 1025-65535. Default: 49. **3.** State—Select to enable this server. Default: Enabled. **4.** Secret—Enter and confirm the 16-digit string for the shared secret used by the TACACS+ client and the server to encrypt and decrypt TACACS+ messages. **5.** Dead time—Enter the time, in seconds, for the quarantine period imposed upon a TACACS+ server that becomes unreachable. Range: 10-10000 seconds. Default: 10. **6.** Authentication methods—Add one or more authentication methods. Default: all.

3. Click **OK**.
4. Save the configuration.

SNMP

This section explains how to configure Simple Network Management Protocol (SNMP) communities and trap receivers. These features are not essential for baseline Oracle Enterprise Communications Broker service, but they are necessary to use an element management system to manage Oracle Enterprise Communications Brokers. They provide important monitoring and system health information that contribute to a robust deployment of the Oracle Enterprise Communications Broker.

Overview

SNMP is used to support monitoring of network-attached devices for conditions that warrant administrative attention. SNMP is comprised of three groups of settings on a Oracle Enterprise Communications Broker. These settings are system-wide configurations including MIB contact information, SNMP community settings, and trap receivers.

Basic SNMP Parameters

The Oracle Enterprise Communications Broker includes several parameters that control basic SNMP functionality. The MIB-related elements are for informational purposes, and are helpful if set. The remainder of the parameters determines if certain Oracle Enterprise Communications Broker events are reported to the SNMP system.

SNMP Community

An SNMP community is a grouping of network devices and management stations used to define where information is sent and accepted. An SNMP device or agent might belong to more than one SNMP community. SNMP communities provide a type of password protection for viewing and setting management information within a community.

SNMP communities also include access level settings. They are used to define the access rights associated with a specific SNMP community. The Oracle Enterprise Communications Broker lets you define two types of access levels: read-only and read-write. You can define multiple SNMP communities on a Oracle Enterprise Communications Broker to segregate access modes per community and NMS host.

Trap Receivers

A trap receiver is an application used to receive, log, and view SNMP traps for monitoring the Oracle Enterprise Communications Broker. An SNMP trap is the notification sent from a network device, the Oracle Enterprise Communications Broker in this case, that declares a change in service. Multiple trap receivers can be defined on a Oracle Enterprise Communications Broker either for redundancy or to segregate alarms with different severity levels to individual trap receivers.

Each server that an element management system is installed on should be configured as a trap receiver on all Oracle Enterprise Communications Broker's managed by that element management system.

SNMP Community Settings

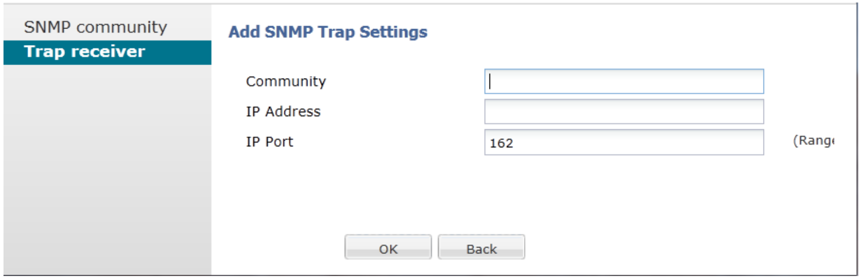
Follow the steps below to configure an SNMP community on your device.

1. Community name—Enter an SNMP community name of an active community where this Oracle Enterprise Communications Broker can send or receive SNMP information. A community name value can also be used as a password to provide authentication, thereby limiting the NMSs that have access to this Oracle Enterprise Communications Broker. With this field, the SNMP agent provides trivial authentication based on the community name that is exchanged in plain text SNMP messages. For example, public. Valid values are alpha-numeric characters. Default is blank.
2. From the SNMP community list, click the Add link. The system displays the Add dialog.
3. IP addresses—Enter an IPv4 address that is valid within this SNMP community. This IPv4 address corresponds with the IPv4 address of the NMS application that monitors or configures this Oracle Enterprise Communications Broker. You can enter multiple addresses, if desired.
4. Click OK to close the Add dialog.

Trap Receiver Settings

Follow the steps below to configure trap receivers on your device.

1. From the Trap receiver list, click the Add link. The system displays the Add SNMP Trap Settings dialog.



Add SNMP Trap Settings

Community
IP Address
IP Port 162 (Rang

2. Community name—Enter the SNMP community name to which this trap receiver belongs. For example, **`Public`**. Valid values are alpha-numeric characters. Default is blank.

3. IP address—Enter the IPv4 address of an authorized NMS. This value is the IPv4 address of an NMS where traps are sent. Enter the IP address in dotted decimal format.
4. IP Port—Enter the port number of an authorized NMS. If you do not specify a port number, the default SNMP trap port of 162 is used.

Web Server Settings

Configure your preferences for the Oracle Enterprise Communications Broker's web server using the Modify web-server-config dialog, available from the Web Server icon. Configuration field descriptions are provided below.

1. Inactivity timeout—Enter the amount of time, in minutes, that the Web GUI must have remained inactive before it ends the Web session. For example, if this timeout value is set as 5, after 5 minutes of no activity, the Web session disconnects. Default is 10. Valid values are 0 to 20. Zero (0) disables this parameter.

   Note: The following HTTP state and HTTPS state parameters may have already been set via the GUI installation wizard on your Oracle Enterprise Communications Broker. You can edit these parameters if required.
2. HTTP state—Specify whether or not to enable HTTP for accessing the Web server. Default is enabled. A check mark indicates enabled, and a blank box indicates disabled.
3. HTTPS state—Specify whether or not to enable HTTPS (secure connection) for accessing the Web server. Default is disabled. A check mark indicates enabled, and a blank box indicates disabled.
4. TLS profile—Enter the Transport Layer Security (TLS) Protocol profile name to use with HTTPS. Valid values are **`alpha-numeric characters.`** Default is blank.

   Note: If you specify a TLS profile, and HTTP is enabled, the Oracle Enterprise Communications Broker checks against the TLS profile table for a match. If there is no match, the applicable errors display during the verification of the configuration.
5. Click OK.

4

Maintenance and Debugging

Oracle Enterprise Communications Broker (ECB) software closely aligns with Oracle Session Border Controller (SBC) software. The vast majority of reference and debugging processes, procedures and information is common across Oracle SBC products.

Generic Maintenance and Debugging Documentation

The following table directs you to other Oracle documentation that provides generic monitoring and debugging information.

Information Type	Documentation
Log File Definition and Descriptions Fault Information Management Manual Configuration Management Process and Procedures	Oracle SBC Maintenance and Troubleshooting Guide
MIB Descriptions MIB Definition and Identification (OID Reference) SNMP GETs SNMP Trap Definition and Descriptions	Oracle SBC MIB Reference Guide
Manual HDR Management HDR Group Definition and Descriptions	Oracle SBC Historical Data Recording (HDR) Resource Guide

Your Oracle Enterprise Communications Broker Image

Your Oracle Enterprise Communications Broker arrives with the most recent, manufacturing-approved run-time image installed on the flash memory. If you want to use this image, you can install your Oracle Enterprise Communications Broker, establish a connection to the Oracle Enterprise Communications Broker, and then begin to configure it. On boot up, your system displays information about certain configurations not being present. You can dismiss these displays and begin configuring your Oracle Enterprise Communications Broker.

If you want to use an image other than the one installed on your Oracle Enterprise Communications Broker when it arrives, you can use the information in this section to obtain and install it.

Obtaining a New Image

You can download software images onto the platform of your Oracle Enterprise Communications Broker from various sources. You can take any one of the following actions:

- Obtain an image from where the Oracle Software Delivery Cloud.
- Obtain an image from your Oracle customer support representative, who will transfer it to your system.

Regardless of how you obtain the image, you need to use Secure File Transfer Protocol (SFTP) to copy it from its source to your Oracle Enterprise Communications Broker.

Upgrade Software - Web GUI System Tab

You can upgrade the system software from the System tab on the Web GUI. The system requires a reboot after the upgrade.

1. From the Web GUI, click the System tab.
2. Click Upgrade Software.
3. Click Verification.
4. Verify that system health, synchronization health, current configuration version, and disk usage are appropriate and adequate for the upgrade.
5. From the drop-down list, select Upload method , and select one of the following methods.
    - Local. Use to select a file from your system for transfer.
    - Flash. Use to select a file already on the device.
    - Network. Use to specify parameters for network boot by way of file transfer.

    The system displays the Upgrade Software dialog with the fields required for your upgrade.
6. Complete the required fields.
    - Software file to upload. (Local) Use Browse to locate the file on your local system.
    - Software file. (Flash) The location and name of the file on the device.
    - Boot file. (Network) The complete name of the boot file.
    - Host IP. (Network) The IP address of the FTP server.
    - FTP username. (Network) The user name to log onto the FTP server.
    - FTP password. (Network) The password to log onto the FTP server.
7. Optional. Select Reboot after upload.
8. Click Complete.
    - If you did not select Reboot after upload, the system displays a message stating that a reboot is required for the changes to take effect.
    - If you selected Reboot after upload, the system displays a message stating that it is about to reboot.
9. Click OK.
    If you selected Reboot after upload, the system reboots.

Displaying Log Files

The Oracle Enterprise Communications Broker allows the user to view log files without having to download them.

1. Click the **System** tab.
    The Oracle Enterprise Communications Broker displays the system navigation panel to the left of the associated controls.

2. Click the **System** tab's **File management** link.
   The Oracle Enterprise Communications Broker displays the **File Management** dialog, which includes **File type** drop down control.
3. Select the **Log** file type.
   The Oracle Enterprise Communications Broker displays file list, displaying all log file categories.
4. Expand a log file category and select a log file by checking the file's check box.
   The Oracle Enterprise Communications Broker enables all applicable command links on the File Management control bar, including the **View** link.
5. Click the **View** link.
   The Oracle Enterprise Communications Broker displays the **Viewing log:[filename]** dialog with the log file's contents. This dialog includes **Close** and **Refresh** buttons.

Displaying System Health

The Oracle Enterprise Communications Broker provides a widget that allows the user to see your device's current health score and state.

1. Click the **Widgets** tab.
   The Oracle Enterprise Communications Broker displays the widget navigation panel to the left of the associated controls.
2. Find and click the **System health** widget group in the **System** widget category.
   The Oracle Enterprise Communications Broker displays the **System health** widget display types, including the link to the **Table** display.
3. Click the **Table** link.
   The Oracle Enterprise Communications Broker displays the **System health** table.
4. Click the Synchronization health button to show extended details on the system's current status.
   The system displays the popup Synchronization health table. The table's information is useful to determine the system's relative ability to act as primary in an HA configuration.

   If the system is deployed in an HA configuration, there is also a Switch over log button, which allows the user to display information about HA switchover events.

Obtaining Support Information

The Oracle Enterprise Communications Broker allows the user obtain a pre-defined file containing information that support personnel normally request.

1. Click the **System** tab.
   The Oracle Enterprise Communications Broker displays the system navigation panel to the left of the associated controls.
2. Click the **System** tab's **Support information** link.
   The Oracle Enterprise Communications Broker displays the **Support Information** dialog, which includes the **Support information** button allowing the user to generate support information output.
3. Click the **Support information** button.
   The Oracle Enterprise Communications Broker displays a **Progress** message box, which indicates the system is generating support information output. When complete, your browser displays a dialog allowing you to decide what to do with the support-info.log file.
4. Follow the dialog's instructions to select the application you want to use to display your support-info.log file or save the file locally.